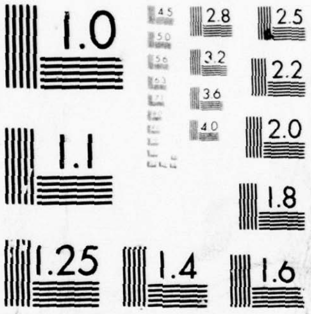
AD-A069 859 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/3
A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS.(U)
AUG 78 J J ALLEMONG DAAB07-72-C-259
UNCLASSIFIED R-818 NL

1 OF 3
AD
A069859



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REPORT R-818 AUGUST, 1978 UILU-ENG 78-2211

CSL COORDINATED SCIENCE LABORATORY

A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS

JOHN JAY ALLEMONG

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. ~~RECIPIENT'S CATALOG NUMBER~~

(9) Doctoral thesis.

4. ~~TITLE (and Subtitle)~~

(6) A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS.

5. ~~TYPE OF REPORT & PERIOD COVERED~~

Technical Report

6. ~~PERFORMING ORG. REPORT NUMBER~~

(14) R-818, UILU-ENG-78-2211

7. AUTHOR(s)

(10) JOHN JAY/ALLEMONG

8. ~~CONTRACT OR GRANT NUMBER(s)~~

(15) DAAB Ø7-72-C-Ø259
✓ NSF-ENG 75-141ØØ

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Joint Services Electronics Program

12. ~~REPORT DATE~~

(11) Aug[redacted]78

13. ~~NUMBER OF PAGES~~

215

14. MONITORING AGENCY NAME & ADDRESS~~(if different from Controlling Office)~~

(12) 223p.

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Power System Dynamics
Singular Perturbations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report applies singular perturbation techniques to linear and non-linear models of a single machine-infinite bus power system and a three machine power system. In the linear realm, we give a method for obtaining state and eigenvalue approximations by computing approximations to the block diagonal system which isolates the fast and slow dynamics. We also give a "growth of model" method for determining the slow and fast variables in the power system models. → next page

097 700 Hu

20. ABSTRACT (continued)

For the non-linear systems, we find that the zero-order approximation of the slow variables is inadequate because it does not account for a non-negligible fast component which is present in these variables. Thus we develop a new and simple method to correct the non-linear zero-order approximations.

UILU-ENG 78-2211

A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS

by

John Jay Allemong

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract DAAB-07-72-C-0259 and in part by the National Science Foundation under Grant NSF ENG 75-14100.

A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS

BY

JOHN JAY ALLEMONG

B.S., University of Illinois, 1973
M.S., University of Illinois, 1974

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1978

Thesis Adviser: Professor M.E. Van Valkenburg

Urbana, Illinois

A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS

John Jay Allemong, Ph.D.
Coordinated Science Laboratory and
Department and Electrical Engineering
University of Illinois at Urbana-Illinois, 1978

This thesis applies singular perturbation techniques to linear and non-linear models of a single machine-infinite bus power system and a three machine power system. In the linear realm, we give a method for obtaining state and eigenvalue approximations by computing approximations to the block diagonal system which isolates the fast and slow dynamics. We also give a "growth of model" method for determining the slow and fast variables in the power system models.

For the non-linear systems, we find that the zero-order approximation of the slow variables is inadequate because it does not account for a non-negligible fast component which is present in these variables. Thus we develop a new and simple method to correct the non-linear zero-order approximations.

ACKNOWLEDGMENT

The author would like to thank his advisor, Professor M. E. Van Valkenburg, for his advice and encouragement during the course of this work. He would also like to express his appreciation to Professor P. V. Kokotovic for his many hours of helpful discussions regarding singular perturbations, power system models, and other matters. He thanks Dr. Joe Chow for his technical discussions and moral support. Finally, he gratefully acknowledges Mary MacMillan and Hazel Corray for their excellent preparation of the manuscript.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Problem Description

This thesis is concerned with the application of singular perturbation techniques to a class of power system dynamics. Generally, power system models are of high dimensionality and are stiff. Therefore, the motivation for this study is clear: since singular perturbation techniques are particularly suitable for analyzing stiff systems of differential equations, we are naturally inclined to ask whether these methodologies are applicable to power system models. This thesis demonstrates that the model describing rotor angle oscillations and associated phenomena can be analyzed by singular perturbation techniques.

The singular perturbation method is often called a model reduction method and it is in the sense that information (e.g. eigenvalues, time responses) about the system is obtained from lower order subsystems. Reduced order modelling has been of interest to power system engineers for many years. Much of the work on reduced order models has been aimed at obtaining dynamic equivalents for transient stability studies.

One of the "classical" methods of equivalencing [1] merely combines machine inertias and parallels their transient reactances. This method requires engineering judgment to determine which machines can be represented by an equivalent machine. The studies in which this method is often used do not include additional dynamics such as voltage regulators and speed governors. For those studies in which the aforementioned dynamics are important, other methods have been devised. One technique (which can be regarded as "classical") is to neglect "small" time constants and other "unimportant" dynamics. Another method [2], which is limited to linear systems, specifies a certain structure for the reduced model and certain eigenvalues of the full model which are to be retained in the reduced model. Then the gains and time constants of the reduced model are calculated. This method requires much computation and a solution does not always exist. In [3] and [4] the authors linearize that portion of the system which is to be replaced by an equivalent. Then the linearized equations are transformed to a diagonal system and certain "fast" modes and "unimportant" modes are neglected. This method requires some prior knowledge of the system in order to determine which portion of the system can be replaced by an equivalent. Also there is some judgment involved in selecting which modes in the equivalent are to be retained and which are to be neglected. Recently, equivalents based on coherency have been investigated [5]. This is a promising avenue of research; although, again, some prior knowledge of the system is needed to determine which machines behave coherently.

Unlike the methods discussed above, the singular perturbation approach does not necessarily generate dynamic equivalents. However, our approach does produce reduced order models in the sense that calculations (of, for example, system time response) are performed on lower order subsystems. Thus the full system is not evaluated directly. A major advantage that the singular perturbation approach has over other reduced order modelling schemes is that the complete system response is recoverable from the subsystem responses. Thus we do not lose information about the system as is the case if some dynamics are completely neglected. Another advantage of singular perturbation methods is that they are not necessarily restricted to linear systems.

1.2 Contributions

Although singular perturbation techniques were applied to a simple power system example in [6], this thesis appears to be the first application of these methodologies to detailed power system models. A necessary part of this study is the development of an iterative scheme to obtain approximations to the block diagonal system which isolates the fast and slow subsystems in a linear singularly perturbed system. In addition, we present a technique (the so called "growth of model" method) for partitioning the power system model into fast and slow subsystems. Finally we give a new and simple method for correcting the zero-order approximation of a non-linear singularly perturbed system in the case when the slow variables contain a non-negligible fast component. All of these methodologies are applied to both a seventh order single machine-infinite bus system and a twentieth order three machine system.

1.3 Review of Power System Dynamics

Power system transients range from travelling wave phenomena on transmission lines to very slow boiler response behavior. In this thesis we will be concerned with synchronous machine rotor angle oscillation transients and associated phenomena. These transients result from changes in the system which cause a momentary mismatch between generation and load. The problem of sustained mismatch between generation and load, although not treated here, has received attention recently [7].

There are basically two types of motion which can result from momentary generation-load mismatch, namely 1) stable motion in which the generator speeds and torque angles move toward the equilibrium corresponding to the new system conditions or 2) unstable motion during which one or more of the generators loses synchronism with the others. The latter situation may result in the removal of generators from the system which in turn could lead to partial or total system collapse due to cascading events.

In order to design and operate the system in a reasonably reliable way, the nature of power system dynamic response must be well understood. This fact was recognized many years ago [1] and the problem of power system "transient" response has received attention ever since.

1.4 Some Aspects of Power System Modelling

In the early studies the system was represented by very simple models. In particular, synchronous machines were represented by a constant voltage behind the transient reactance [1] and the dynamics of voltage regulators and governors were generally neglected.

In recent years it has become apparent that these simple models are not adequate to describe some kinds of power system dynamic response. For example, sustained low frequency power oscillations which are apparently associated with voltage regulator action have been observed [8]. The need to investigate these phenomena combined with the availability of large, high speed digital computers have resulted in a vast increase in the complexity of power system models [9].

Unfortunately there is a price to be paid for using more accurate models of power system components. First, the dimension of the system of differential equations describing a large power system is very large. Second, the complexity of the differential equations can often obscure some underlying physical phenomena which essentially determine the system behavior. Finally, it is well known that the complete set of differential equations is stiff [9]; that is, the responses consist of some slow and fast modes.

This stiffness leads to serious difficulties in carrying out numerical integration. The singular perturbation approach alleviates both the dimensionality and stiffness problems by analyzing the fast and slow subsystems separately. Hence, singular perturbation methods appear to be particularly suitable for studying power system models.

1.5 Chapter Preview

In Chapter 2 we discuss some basic principles of singular perturbations. We then present a method for obtaining an approximation to the block diagonal system which isolates the fast and slow dynamics in a linear system. Finally we discuss the problem of partitioning an arbitrary system into its fast and slow subsystems and we give a procedure for partitioning our power system model.

Chapter 3 considers a single machine-infinite bus system. First the partitioning of this system is determined by the growth of model technique described in Chapter 2. Next the complete linear model is analyzed by the iterative scheme introduced in Chapter 2 and both eigenvalue and time response approximations are found to be quite good. Finally the nature of the zero-order approximation of the non-linear system is explored. This approximation is found to exhibit some deficiences. A method to correct them is proposed and verified.

In Chapter 4 we repeat the analysis of Chapter 3 for a multi-machine system. We find that by introducing a change in angle-speed variables and by applying the experience gained with the single machine system, the partitioning of this system is straight-forward. State and eigenvalue approximations of the linear model are seen to be quite accurate but more iterations are required to achieve this accuracy than in the single machine case. Finally the twentieth order non-linear system is analyzed and the zero-order approximation is found to suffer from the same type of difficulty as the zero-order approximation of the single machine system. The correction procedure is applied and the approximations are seen to be improved relative to the zero-order approximation.

2. SINGULAR PERTURBATION METHODS

2.1 Introduction

Singular perturbation methods [10] are applicable to systems of differential equations of the following form

$$\dot{x} = f(x,z,u),\ x(0) = x_0 \tag{2.1a}$$

$$\mu\dot{z} = g(x,z,u),\ z(0) = z_0 \tag{2.1b}$$

where x is an n-vector, z is an m-vector, u is the input vector of dimension r and μ is a small positive parameter known as the singular perturbation parameter. The presence of μ indicates that there are fast dynamics in x and z. Setting $\mu = 0$ in (2.1b) is equivalent to neglecting the fast phenomena in the response and gives the following slow model

$$\dot{\bar{x}} = f(\bar{x},\bar{z},\bar{u}),\ \bar{x}(0) = x_0 \tag{2.2a}$$

$$0 = g(\bar{x},\bar{z},\bar{u}) \tag{2.2b}$$

where the overbar indicates the slow part of these quantities. The fast part of the response can be approximately recovered by solving the system

$$\mu\dot{\tilde{z}} = g(\bar{x},\ \bar{z}+\tilde{z},\ \bar{u}+\tilde{u}),\ \tilde{z}(0) = z_0 - \bar{z}(0)\ . \tag{2.3}$$

In (2.3) the tilde denotes the fast part of these quantities. Notice that the approximate responses, $\bar{x}$ and $\bar{z}+\tilde{z}$ are obtained by solving lower order subsystems thus reducing the number of differential equations which have to be integrated simultaneously. In addition, the fast and slow subsystems are solved separately in their own time scales, thereby ameliorating the problem of stiffness.

The linear time-invariant version of (2.1) is

$$\dot{x} = Ax + Bz + Gu,\ x(0) = x_0 \tag{2.4a}$$

$$\mu\dot{z} = Cx + Dz + Hu,\ z(0) = z_0\quad . \tag{2.4b}$$

In this case, the usual slow subsystem model is

$$\dot{\overline{x}} = (A-BD^{-1}C)\overline{x} + (G-BD^{-1}H)\overline{u}, \quad \overline{x}(0) = x_0 \tag{2.5a}$$

$$\overline{z} = -D^{-1}C\overline{x} - D^{-1}H\overline{u} \quad , \tag{2.5b}$$

and the fast subsystem model is

$$\mu\dot{\tilde{z}} = D\tilde{z} + H\tilde{u}, \quad \tilde{z}(0) = z_0 - \overline{z}(0) \ . \tag{2.6}$$

In either the linear or non-linear version described above, the responses x and z are approximated by $\overline{x}$ and $\overline{z}+\tilde{z}$. Therefore, this approximation does not account for a possible fast component in x. In many cases this approximation is satisfactory. However, if the system is severely disturbed, the fast response in x may be large. In this case, the effect of the fast variables on the slow subsystem may not be negligible. Then it is generally necessary to use modified procedures to calculate the approximate responses. In the linear case, one such method is to obtain an approximation to the block diagonal system which isolates the fast and slow subsystems by using the Chang transformation [11]. Another method to arrive at the same result will be described in the next section. In the non-linear case, the method used depends on the specific non-linear functions in (2.1). In Chapters 3 and 4 we give one method for calculating corrections to the basic approximations for the non-linear equations encountered in the power system models presented there.

Finally, the singular perturbation parameter μ is not always identifiable, nor is it necessary that it be isolated. The only requirement for using singular perturbation techniques to analyze a system is that the system must possess the multiple time scale property. Once this property has been established for the system, then it is necessary that the system be partitioned correctly. We will return to this topic later in this chapter.

2.2 An Iterative Scheme

One way to obtain accurate fast and slow approximations is to carry out the calculations on a block diagonal system which isolates the fast and slow subsystems. We now give a method to construct, approximately, this block diagonal system. Consider the linear, time-invariant system (2.4) with u = 0 for simplicity

$$\dot{x} = Ax + Bz, \quad x(0) = x_0 \tag{2.7a}$$

$$\mu\dot{z} = Cx + Dz, \quad z(0) = z_0 \ . \tag{2.7b}$$

Define

$$\eta_1 = z + D^{-1}Cx \ . \tag{2.8}$$

We substitute (2.8) into (2.7a) to give

$$\begin{aligned}\dot{x} &= (A-BD^{-1}C)x + B\eta_1 \\ &= A_1 + B\eta_1 \ .\end{aligned} \tag{2.9}$$

Now we take the time derivative on both sides of (2.8) and substitute (2.7b) and (2.9) to yield

$$\begin{aligned}\mu\dot{\eta}_1 &= \mu D^{-1}C\ A_1x + (D + \mu D^{-1}CB)\eta_1 \\ &= C_1x + D_1\eta_1 \ .\end{aligned} \tag{2.10}$$

This procedure can be repeated iteratively. At the end of the k-th iteration, the result is

$$\dot{x} = A_kx + B\eta_K \tag{2.11a}$$

$$\mu\dot{\eta}_k = C_kx + D_k\eta_k \tag{2.11b}$$

where

$$A_k = A_{k-1} - BD_{k-1}^{-1}C_{k-1} \tag{2.12a}$$

$$C_k = \mu D_{k-1}^{-1}\ C_{k-1}A_k \tag{2.12b}$$

$$D_k = D_{k-1} + \mu D_{k-1}^{-1}\ C_{k-1}B\quad . \tag{2.12c}$$

In order to avoid computing D_{k-1}^{-1} at each iteration, we can approximate these inverses. Let $G = G_0 + \mu G_1$. Then

$$\begin{aligned} G^{-1} &= (G_0 + \mu G_1)^{-1} \\ &= M_0 + \mu M_1 + \frac{1}{2}\mu^2 M_2 + \cdots \quad . \end{aligned} \tag{2.13}$$

It is clear that $M_0 = G_0^{-1}$. To obtain the higher order terms, we employ repeated differentiation. For M_1 we have

$$M_1 = \frac{d}{d\mu}(G_0 + \mu G_1)^{-1}\Big|_{\mu=0} \quad . \tag{2.14}$$

But for any non-singular matrix A, $AA^{-1} = I$. Hence

$$\frac{dA}{d\mu}A^{-1} + A\frac{dA^{-1}}{d\mu} = 0$$

or

$$\frac{dA^{-1}}{d\mu} = -A^{-1}\frac{dA}{d\mu}A^{-1} \quad . \tag{2.15}$$

Applying (2.15) to (2.14) we obtain for M_1

$$M_1 = -G_0^{-1}G_1G_0^{-1} \ . \tag{2.16}$$

Repeating this procedure for other higher order terms, we find that the expansion for G^{-1} can be written in the following form

$$G^{-1} = G_0^{-1}\{I-\mu G_1G_0^{-1}[I-\mu G_1G_0^{-1}\ (I-\ldots)]\} \ . \tag{2.17}$$

For example, an approximation of D_1^{-1} is

$$D_1^{-1} \cong D^{-1}\{I-\mu D^{-1}CBD^{-1}[I-\mu D^{-1}CBD^{-1}(I-\mu D^{-1}CBD^{-1})]\} \ . \tag{2.18}$$

Notice that the expression for C_k contains μ as a multiplying factor. Hence, at the end of each iteration, the elements of C_k should be smaller than the elements of C_{k-1}. Thus, after some convenient number of iterations, say k, the coupling term $C_k x$ can be dropped. Then the fast subsystem is defined as

$$\mu\dot{\eta} = \mathcal{D}\eta \ . \tag{2.19}$$

Now we write (2.11a) as

$$\dot{x} = \mathcal{A}x + B\eta \ . \tag{2.20}$$

To decouple the slow subsystem from the fast subsystem, we substitute $\eta = \mu\mathcal{D}^{-1}\dot{\eta}$ into (2.20). The result can be written as

$$\dot{x} - \mu B\mathcal{D}^{-1}\dot{\eta} = \mathcal{A}x \tag{2.21}$$

which suggests the definition of a new slow variable

$$\xi_1 = x - \mu B\mathcal{D}^{-1}\eta \ . \tag{2.22}$$

With this definition, (2.21) becomes

$$\begin{aligned}\dot{\xi}_1 &= \mathcal{A}\xi_1 + \mu\mathcal{A}B\mathcal{D}^{-1}\eta \\ &= \mathcal{A}\xi_1 + B_1\eta \ .\end{aligned} \tag{2.23}$$

This scheme can be employed iteratively. After the k-th iteration, the result is

$$\dot{\xi}_k = \mathcal{A}\xi_k + B_k\eta \tag{2.24}$$

where

$$B_k = \mu\mathcal{A}B_{k-1}\mathcal{D}^{-1} \ . \tag{2.25}$$

As in the case of C_k, the elements of B_k are reduced at each iteration. Hence, after some suitable number of iterations, the coupling terms can be dropped and the slow subsystem defined as

$$\dot{\xi} = \mathcal{A}\xi \ . \tag{2.26}$$

The number of iterations performed on the η subsystem does not necessarily have to be equal to the number performed on the ξ subsystem. However, for computational symmetry, we will always carry out the same number of iterations on both subsystems.

We now turn our attention to the problem of recovering x and z from ξ and η. First, we have

$$\eta_k = \eta_{k-1} + D^{-1}_{k-1} C_{k-1} x \quad . \tag{2.27}$$

If (2.27) is summed from 1 to k, the result is

$$\sum_{m=1}^{k} \eta_m = \sum_{m=1}^{k} \eta_{m-1} + \sum_{m=1}^{k} D_{m-1} C_{m-1} x \tag{2.28}$$

or

$$\eta_k = \eta_0 + \sum_{m=1}^{k} D^{-1}_{m-1} C_{m-1} x \quad . \tag{2.29}$$

Let

$$L = \sum_{m=1}^{k} D^{-1}_{m-1} C_{m-1} \quad . \tag{2.30}$$

Then (2.29) can be rewritten as

$$\eta = z + Lx \quad . \tag{2.31}$$

In (2.31) we have dropped the subscript on η to signify that the result of the last iteration is taken to be the true fast response.

Similarly, for ξ we have

$$\xi_k = \xi_{k-1} - \mu B_{k-1} \mathcal{D}^{-1} \eta \tag{2.32}$$

$$\sum_{m=1}^{k} \xi_m = \sum_{m=1}^{k} \xi_{m-1} - \mu \sum_{m=1}^{k} B_{m-1} \mathcal{D}^{-1} \eta \tag{2.33}$$

$$\xi_k = \xi_0 - \mu \sum_{m=1}^{k} B_{m-1} \mathcal{D}^{-1} \eta \quad . \tag{2.34}$$

Let

$$H = \sum_{m=1}^{k} B_{m-1}\partial^{-1} \quad . \tag{2.35}$$

Thus, (2.34) becomes

$$\xi = x - \mu H\eta \quad . \tag{2.36}$$

By substituting (2.31) into (2.36) and simplifying the result, we arrive at the following transformation of variables

$$\begin{bmatrix} \xi \\ \\ \eta \end{bmatrix} = \begin{bmatrix} I-\mu HL & -\mu H \\ & \\ L & I \end{bmatrix} \begin{bmatrix} x \\ \\ z \end{bmatrix} \quad . \tag{2.37}$$

Incidently, this transformation gives us a convenient way to compute the initial conditions for simulating the ξ and η subsystems. The inverse transformation is

$$\begin{bmatrix} x \\ \\ z \end{bmatrix} = \begin{bmatrix} I & \mu H \\ & \\ -L & I-\mu LH \end{bmatrix} \begin{bmatrix} \xi \\ \\ \eta \end{bmatrix} \quad . \tag{2.38}$$

This transformation is a special case of the transformation used by Chang [11]; it is also obtained in [12].

The approximation procedure which has been outlined in this section is somewhat different than the basic approximation scheme presented in the previous section. Here, we compute approximations to the block diagonal system which isolates the fast and slow subsystems. Then, the original variables are reconstructed by means of the transformation (2.38). The calculations are always performed on lower order subsystems.

2.3 Use of Singular Perturbations in Analyzing Power System Dynamics

In later chapters we use a power system model which is valid for studying synchronous machine behavior for a period of several seconds following an initiating disturbance. This model consists of swing equations, flux decay dynamics, and voltage regulator representation. It possesses the multiple time scale property since it encompasses dynamics ranging from fast control loops to system frequency drift (in the multi-machine case). Therefore, singular perturbation techniques are appropriate for studying this power system model. By means of singular perturbations, we can obtain reduced order models which alleviate the stiffness and dimensionality problems of the full order models.

2.4 Techniques of Partitioning the System

One weakness of the singular perturbation approach is that there is no general method for locating the fast and slow variables in an arbitrary system. Fortunately the model used in this thesis allows us to determine the partitioning fairly simply. We shall call the technique about to be described the "growth of model" method.

First, all the synchronous machines in the system are represented by only their swing equations and the eigenvalues of the linearized system are computed. Next, flux decay dynamics are included with the swing equations. The eigenvalues of the linearized system are again computed. It is an interesting property of these first two models that the flux modes are weakly coupled to the swing modes. Hence, in the flux decay model, it is possible to immediately recognize the flux modes and the swing modes. This behavior is not coincidental since the swing modes are determined primarily by the machines' moments of inertia and the admittances of the interconnecting network. The inclusion of flux dynamics introduces an energy dissipation mechanism which somewhat affects the damping of the swing modes.

Finally the voltage regulators are included with the model and the eigenvalues of the linearized system are examined. The swing modes and the mode due to quadrature axis flux dynamics are discernable. The other eigenvalues produced by this model result from some interactions among the variables. The origins of these eigenvalues are clarified after these interactions are understood.

Thus by the growth of model method we can identify which variables are primarily fast and which are primarily slow. This allows the system to be partitioned into the form of (2.1) or (2.4).

There is another method for partitioning a system when the system can be conveniently represented by a block diagram. The block diagram is examined for small time constants and loops containing large gains since their presence gives rise to fast dynamics. Hence, the fast and slow variables can often be identified by inspection of the block diagram. We will use this method in the next chapter as an aid in classifying the dynamics of the voltage regulator model.

3. SINGLE MACHINE CASE STUDY

3.1 Introduction

In this chapter we analyze a single machine-infinite bus system by singular perturbation methods. Both linear and non-linear results are presented. Although this system has no true counterpart in practice, many practical systems can be analyzed by approximating them as single machine-infinite bus systems. In addition the multi-machine system considered in the next chapter retains many of the characteristics of the single machine system. Hence, it is beneficial to analyze and understand the simpler system before trying to analyze the multi-machine system.

3.2 System Description

The single machine-infinite bus system is shown in Fig. 3.1. The voltage regulator model used in this study is the IEEE Type 1 representation [13] a block diagram of which is shown in Fig. 3.2. In the model used here saturation non-linearity, represented on the block diagram by S_E, is retained but limit type non-linearities are neglected.

Following are the differential equations for each of the three models. The swing model equations are well known [1]. The full model equations are derived in Appendix A and the flux decay model equations are obtained from the former by holding the field voltage (E_{fd}) constant and dropping the voltage regulator equations.

Swing Model

$$\dot{\delta} = 377\ (\Omega - 1) \tag{3.1a}$$

$$\dot{\Omega} = \frac{1}{2H}\left[P_{in} - D(\Omega - 1) - YE'V_i \sin\delta\right] \tag{3.1b}$$

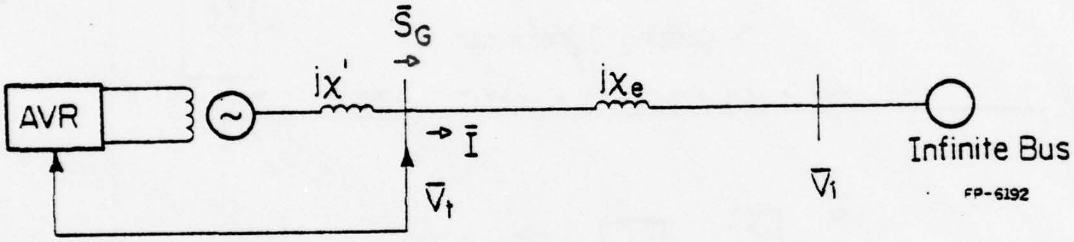


Fig. 3.1. Single machine-infinite bus system.

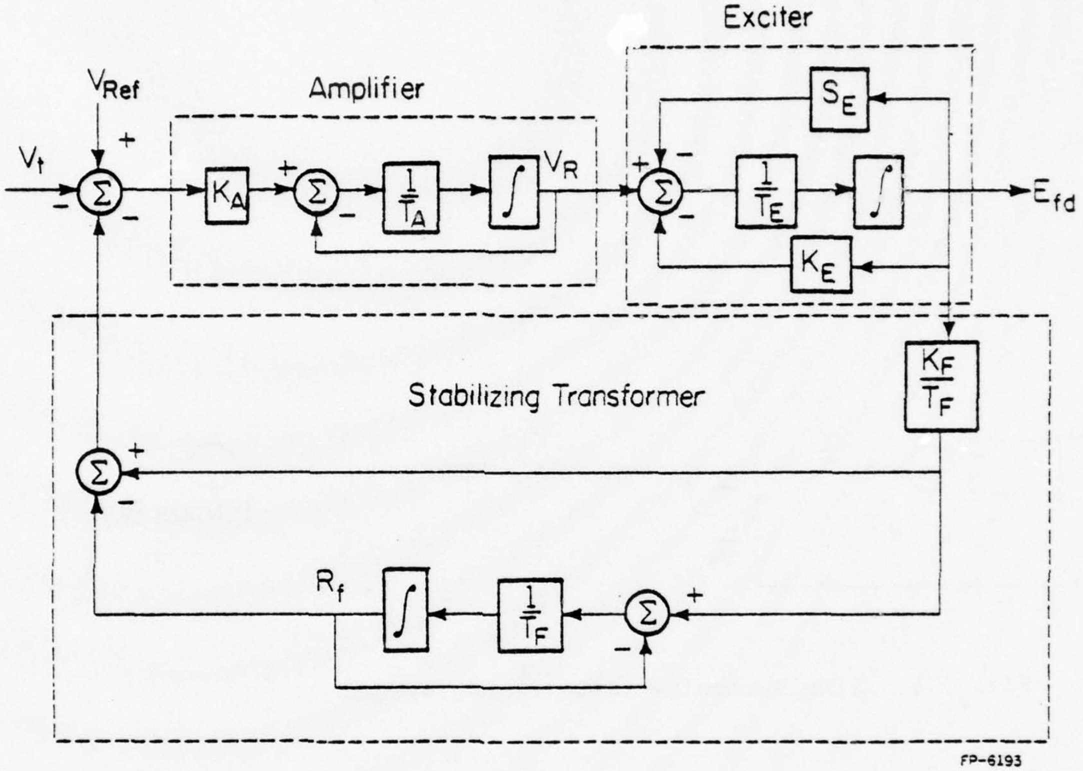


Fig. 3.2. IEEE type 1 voltage regulator.

Flux Decay Model

$$\dot{e}_q' = \frac{1}{T_{d_0}'}\{ - [1 + (x_d - x')Y]\ e_q' - (x_d - x')YV_i \sin\delta + E_{fd}\} \tag{3.2a}$$

$$\dot{e}_d' = \frac{1}{T_{q_0}'}\{ (x_q - x')YV_i \cos\delta \ - \ [1 + (x_q - x')Y]\ e_d'\} \tag{3.2b}$$

$$\dot{\delta} = 377\ (\Omega - 1) \tag{3.2c}$$

$$\dot{\Omega} = \frac{1}{2H}\ [\ \frac{P_{in}}{\Omega} - D(\Omega-1) - YV_i (e_q' \cos\delta + e_d' \sin\delta)] \tag{3.2d}$$

Full Model

$$\dot{e}_q' = \frac{1}{T_{d_0}'}\{ - [1 + (x_d - x')Y]\ e_q' - (x_d - x')YV_i \sin\delta + E_{fd}\} \tag{3.3a}$$

$$\dot{R}_f = \frac{1}{T_F}\ (-R_f + \frac{K_F}{T_F}\ E_{fd}) \tag{3.3b}$$

$$\dot{e}_d' = \frac{1}{T_{q_0}'}\{ (x_q - x')Y V_i \cos\delta - [1 + (x_q - x')Y]\ e_d'\} \tag{3.3c}$$

$$\dot{\delta} = 377\ (\Omega - 1) \tag{3.3d}$$

$$\dot{\Omega} = \frac{1}{2H}\ [\ \frac{P_{in}}{\Omega} - D(\Omega-1) - YV_i (e_q' \cos\delta + e_d' \sin\delta)] \tag{3.3e}$$

$$\dot{E}_{fd} = \frac{1}{T_E}\ \{ - [K_E + S_E(E_{fd})]\ E_{fd} + V_R\} \tag{3.3f}$$

$$\dot{V}_R = \frac{1}{T_A}\ [K_A(R_f - \frac{K_F}{T_F}\ E_{fd} - V_t + V_{Ref}) - V_R] \tag{3.3g}$$

$$V_t = [(1 - x'Y)^2 (e_q'^2 + e_d'^2) + 2(1 - x'Y)(x'YV_i)(e_d' \cos\delta - e_q' \sin\delta) + (x'YV_i)^2]^{1/2} \tag{3.3h}$$

$$S_E(E_{fd}) = A_{sat} \exp[(B_{sat})(E_{fd})] \tag{3.3g}$$

The parameters used in this study are

$$\begin{array}{ll}
H = 5.0 \text{ sec} & T_A = 0.06 \text{ sec} \\
D = 2.0 \text{ pu} & T_E = 0.5 \text{ sec} \\
x_d = 1.2 \text{ pu} & T_F = 1.0 \text{ sec} \\
x_q = 1.0 \text{ pu} & K_A = 25 \\
x' = 0.25 \text{ pu} & K_E = -0.0445 \\
x_e = 0.25 \text{ pu} & K_F = 0.16 \\
T'_{d_0} = 5.0 \text{ sec} & A_{sat} = 0.001123 \\
T'_{q_0} = 0.50 \text{ sec} & B_{sat} = 0.3043 \quad .
\end{array}$$

3.3 Determination of Partitioning

To determine the fast and slow variables in the full system, we use the growth of model and block diagram methods discussed in the last chapter. The linearized equations for each of the models described in the last section can be written in the following form

$$\dot{w} = Aw \quad . \tag{3.4}$$

In each case the terminal conditions about which the linearization is performed are

$$\begin{aligned}
\overline{V}_i &= 1.0 + j0 \text{ pu} \\
\overline{V}_t &= 1.0 + j0.2 \text{ pu} \\
\overline{S}_g &= 0.8 + j0.1608 \text{ pu.}
\end{aligned}$$

Swing Model

In the swing model

$$w = [\Delta\delta \;\; \Delta\Omega]' \tag{3.5}$$

(' denotes "transpose"). The A matrix is

$$\begin{bmatrix} 0 & 377 \\ -0.2 & -0.2 \end{bmatrix}$$

the eigenvalues of which are

$$-0.1 \pm j8.68 \quad .$$

Since this model accounts for only the mechanical motion of the rotor, this mode is clearly identifiable as the rotor oscillatory mode. Note that the "frequency" of this mode (1.38 Hz) is quite typical of values found in practice.

Flux Decay Model

In the flux decay model

$$w = [\Delta e_q' \ \Delta e_d' \ \Delta\delta \ \Delta\Omega]' \tag{3.6}$$

and the A matrix is

$$\begin{bmatrix} -0.58 & 0 & -0.269 & 0 \\ 0 & -5.0 & 2.12 & 0 \\ 0 & 0 & 0 & 377 \\ -0.141 & 0.142 & -0.2 & -0.28 \end{bmatrix} \quad .$$

The eigenvalues of this matrix are

$$-0.887 \pm j8.41$$

$$-3.78$$

$$-0.301 \quad .$$

The mechanical mode is clearly visible (-0.887 ± j8.41). The imaginary part of this mode has not been affected very much by the inclusion of flux decay dynamics; but the damping of this mode has been increased substantially. The energy dissipation provided by the rotor circuits accounts for this additional damping.

Flux decay dynamics are responsible for the remaining two modes. The root at -3.78 corresponds to $\Delta e'_d$ while the root at -0.301 corresponds to $\Delta e'_q$. This determination is obvious since $T'_{q_0} = 0.5$ sec. and $T'_{d_0} = 5.0$ sec.

The quadrature axis rotor coil has a smaller time constant than the direct axis rotor coil (i.e. it is "faster"). Thus, we might suspect that more of the mechanical mode damping is accounted for by the presence of the quadrature axis coil than by the presence of the direct axis coil. If we modify this model's A matrix by removing the dependence of $\Delta\dot{\Omega}$ on $\Delta e'_d$ we obtain the following modified A matrix

$$\begin{bmatrix} -0.58 & 0 & -0.269 & 0 \\ 0 & -5.0 & 2.12 & 0 \\ 0 & 0 & 0 & 377 \\ -0.141 & 0 & -0.2 & -0.28 \end{bmatrix} .$$

The eigenvalues of this matrix are

$-0.235 \pm j8.68$

-5.0

-0.390 .

Hence, in this case, a sizeable portion of the mechanical mode damping is attributable to the quadrature axis coil.

<u>Full Model</u>

In the full model

$$w = [\Delta e'_q \; \Delta R_f \; \Delta e'_d \; \Delta\delta \; \Delta\Omega \; \Delta E_{fd} \; \Delta V_R]' \qquad (3.7)$$

and the A matrix is

$$\begin{bmatrix} -0.58 & 0 & 0 & -0.269 & 0 & 0.2 & 0 \\ 0 & -1.0 & 0 & 0 & 0 & 0.16 & 0 \\ 0 & 0 & -5.0 & 2.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 377 & 0 & 0 \\ -0.141 & 0 & 0.141 & -0.2 & -0.28 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0838 & 2.0 \\ -173 & 417 & -116 & 40.9 & 0 & -66.7 & -16.7 \end{bmatrix}$$

the eigenvalues of which are

$-0.861 \pm j8.39$

-3.93

$-0.362 \pm j0.558$

$-8.53 \pm j8.22$.

The mechanical mode is clearly present ($-0.861 \pm j8.39$) as is the mode due to $\Delta e_d'$ (-3.93). However, the origin of the other four modes is somewhat obscure.

To understand how these other modes arise, we make use of the block diagram reduction technique alluded to in Chapter 2. We consider the voltage regulator model of Fig. 3.2. The block diagram of the linearized equations correponding to the loop consisting of the amplifier, the exciter, and the K_F/T_F feedback path can be drawn in the form of Fig. 3.3. The characteristic equation of this subsystem is

$$s^2 + (\frac{1}{T_1} - \frac{1}{T_2})s + \frac{K_1K_2K_3 - 1}{T_1T_2} \quad . \tag{3.8}$$

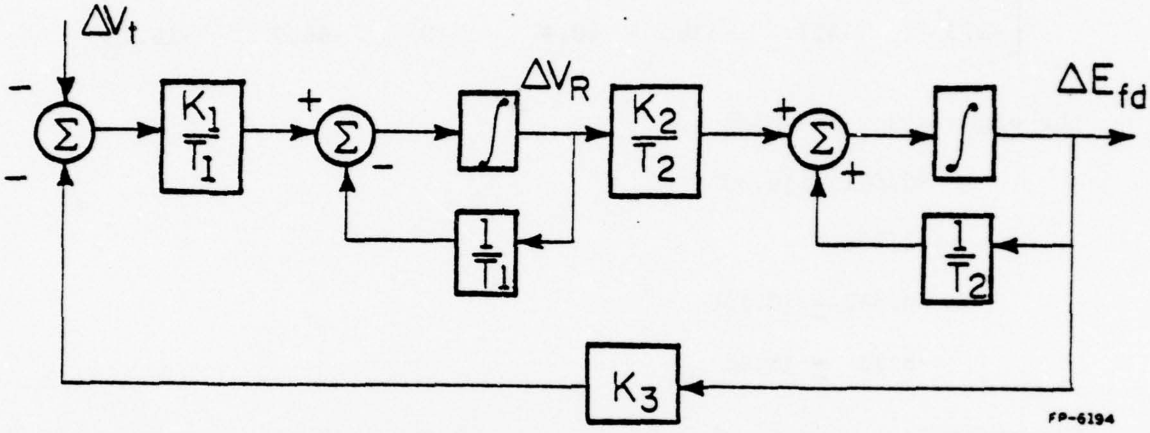


Fig. 3.3. Equivalent block diagram of inner loop of voltage regulator model.

The gain K_1 is usually large for two reasons. First, the voltage regulator is a proportional controller; hence large gain is needed to produce small steady state error. Second, large gain makes the regulator faster. The effect of increasing gain is to move the roots of (3.8) along the asymptotes shown in Fig. 3.4. Hence, we can consider this subsystem as singularly perturbed in the sense of high frequency oscillations [6]. For the numerical values in the system matrix, the roots are $-8.29 \pm j7.95$. Thus the fast pair $(-8.53 \pm j8.22)$ is attributable to the aforementioned loop.

To understand the origin of the remaining complex pair, we consider the simpler system of an open circuited synchronous machine with voltage regulator. If the inner voltage regulator loop is reduced by singular perturbation, the resulting slow subsystem model is as shown in Fig. 3.5. The characteristic equation of this subsystem is

$$s^2 + 1.45\ s + 1.26 \tag{3.9}$$

the roots of which are $-0.725 \pm j0.857$. Thus, we can conclude that in the full model, the complex pair $-0.362 \pm j0.558$ arises from the interaction of $\Delta e_q'$ and ΔR_f.

Although the partitioning of the full system into fast and slow subsystems is not unique, one possibility is to place $-0.362 \pm j0.558$ in the slow subsystem and all other modes in the fast subsystem. From the previous discussions, we see that we should place $\Delta e_q'$ and ΔR_f in the slow subsystem and the other variables in the fast subsystem.

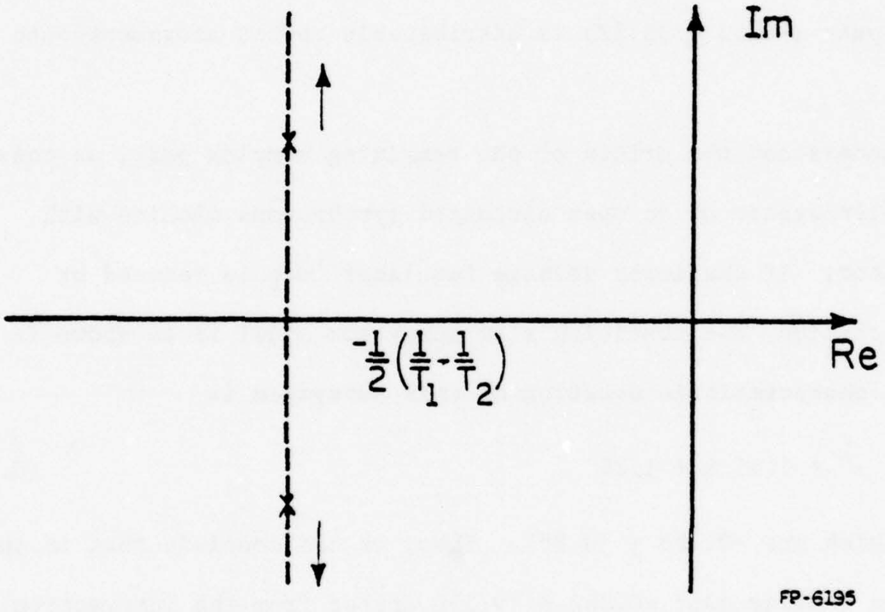


Fig. 3.4. Effect of increasing gain on the roots of Eq. (3.8).

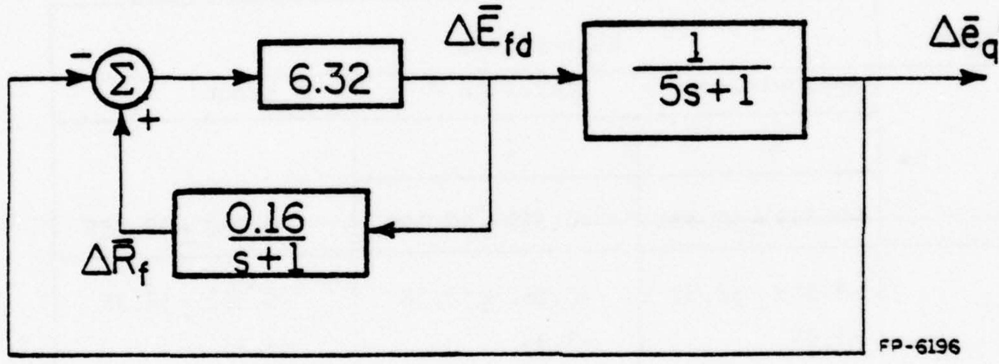


Fig. 3.5. Slow subsystem model of open-circuited synchronous machine with voltage regulator.

3.4 Analysis of Linearized System

In this section we analyze the seventh order linear system by means of the iterative singular perturbation scheme presented in Chapter 2. First we examine the eigenvalue approximations for the seventh order system matrix given previously. Table 1 shows approximate and exact eigenvalues. The approximate eigenvalues are computed from the subsystem matrices after the specified number of iterations.

Table 1. Eigenvalue Approximations for Seventh Order System

Subsystem	Eigenvalues		
	Approximate - Iteration #		Exact
	1	2	
Slow	-0.399 ± j0.581	-0.359 ± j0.560	-0.362 ± j0.558
Fast	-0.858 ± j8.39 -3.94 -8.49 ± j8.22	-0.862 ± j8.38 -3.94 -8.53 ± j8.22	-0.861 ± j8.39 -3.93 -8.53 ± j8.22

Note that the usual zero-order approximation is not used here since it is not consistent with the iterative scheme.

After the first iteration, the fast eigenvalues are approximated quite well but the slow subsystem eigenvalues are somewhat inaccurate. After the second iteration, all of the eigenvalues are approximated very closely.

A very important test of the subsystem approximation scheme is the accuracy of the approximate responses, $\hat{x}(t)$ and $\hat{z}(t)$. To test this aspect of the methodology, we simulate a system disturbance. The disturbance chosen for this study is a 5 cycle, 3 phase stub fault applied at the infinite bus. Figures 3.6 - 3.12 show the simulation results for a period of 10 seconds following the disturbance. In each case the solid curve is the exact solution and the dotted curve is the approximate solution obtained by using one iteration of the iterative scheme. The initial conditions were calculated by integrating the non-linear equations (3.3) during the fault and subtracting from the variables' values at fault clearing the corresponding equilibrium values.

Figures 3.6 and 3.7 are the slow variables. The predominant slow behavior of these responses is apparent. However, in both cases, an initial fast component is clearly present. This fast component is probably due to the forcing effect of the field voltage (note that the field voltage, E_{fd}, appears in the equations for e'_q (3.3a) and R_f (3.3b)) which attains a relatively large value during the fault and shortly after the fault is cleared. Once the field voltage has decreased to nearly its equilibrium value, the slow variables regain their dominant slow behavior. The approximation method used here takes into account a possible fast component in the slow variables since $\hat{x} = \xi + \mu H\eta$. This relationship also shows that a large initial condition in the fast variables has a non-negligible effect on the slow subsystem initial condition because $\xi(0) = \hat{x}(0) - \mu H\eta(0)$.

Even though there is some error in the approximation of the slow variables, it is clear that with only one iteration, the slow variables are approximated quite well. The curves of the exact and approximate fast variables, Figs. 3.8 - 3.12, are in general indistinguishable.

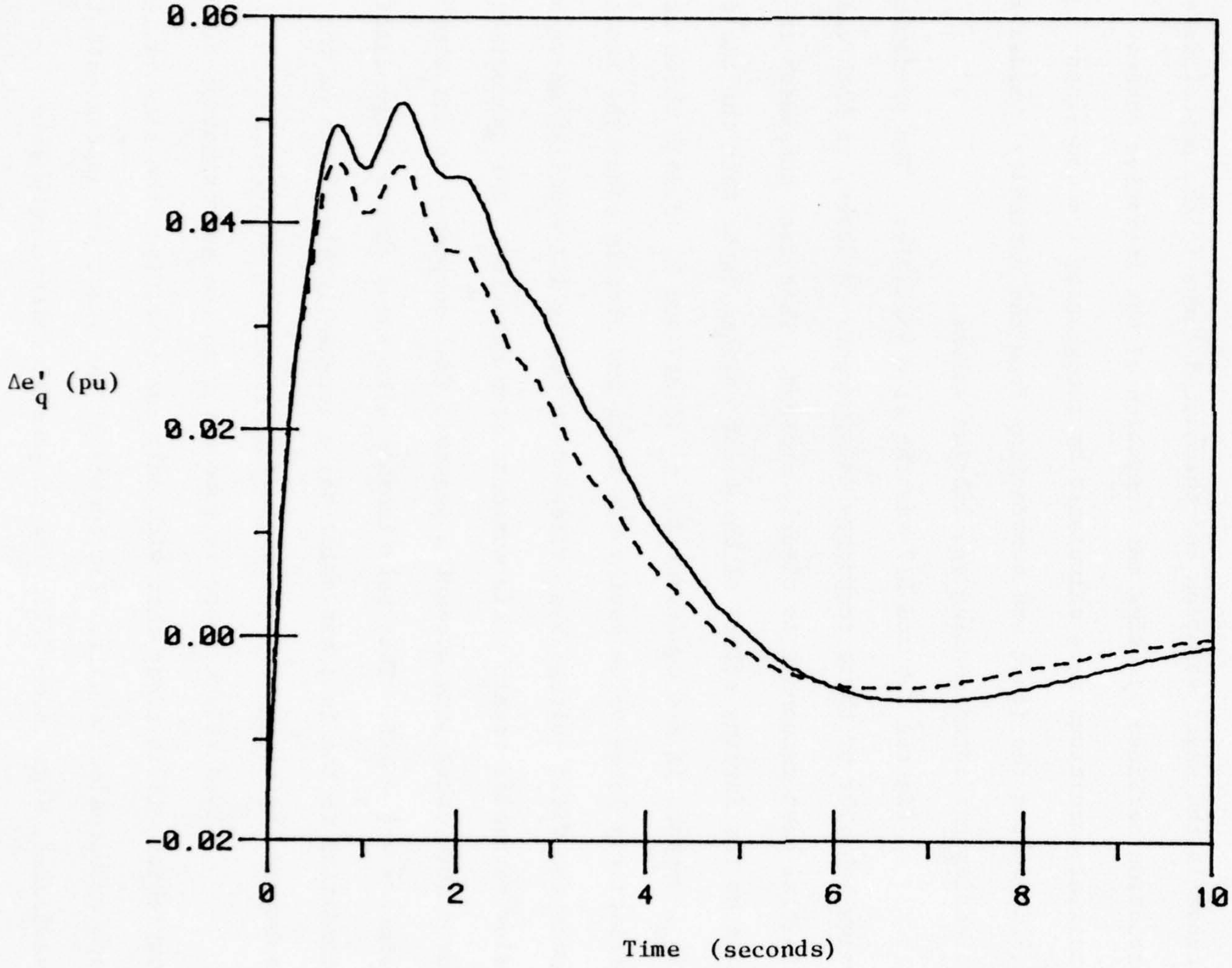


Fig. 3.6. Plots of exact and approximate $\Delta e_q'$.

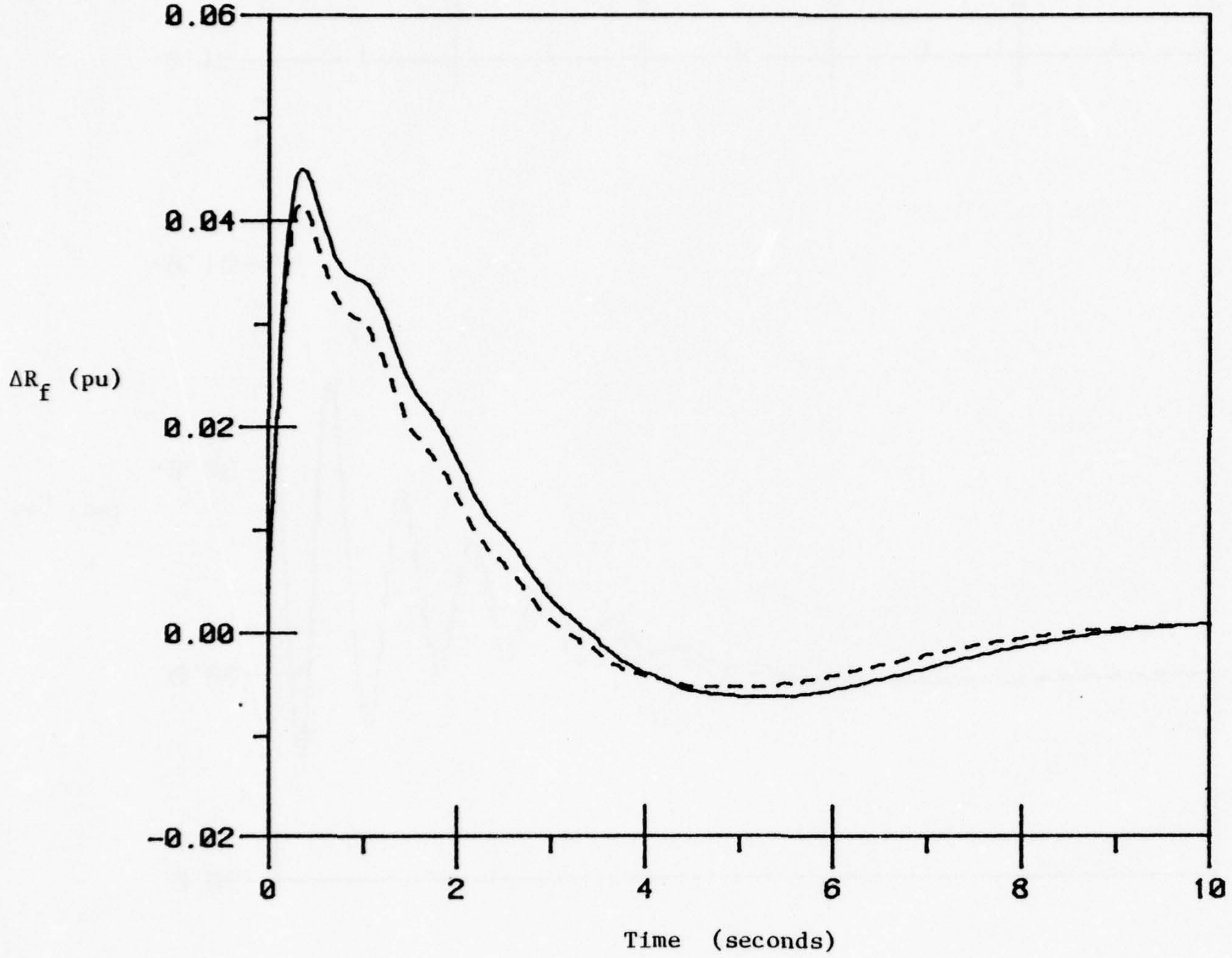


Fig. 3.7. Plots of exact and approximate ΔR_f.

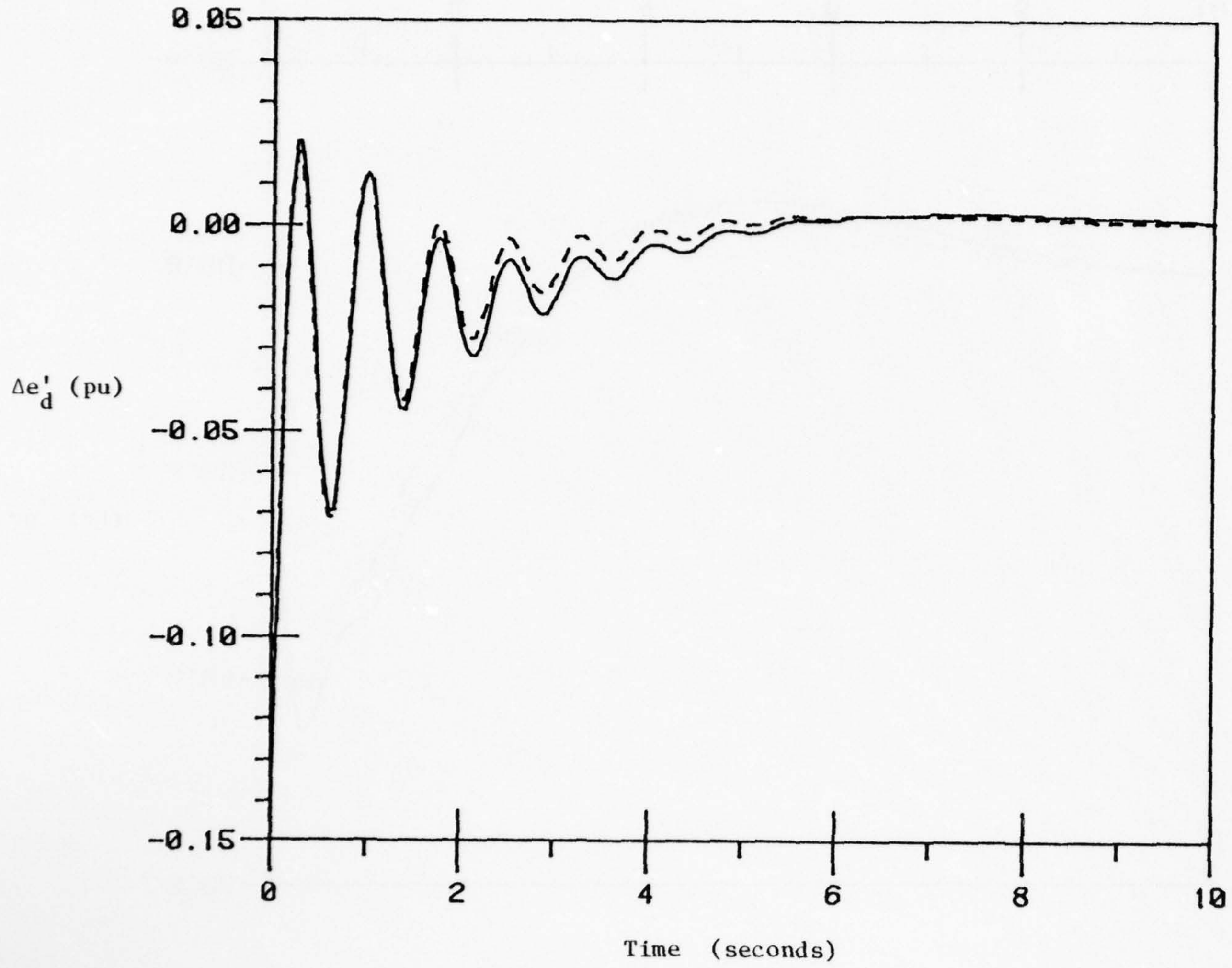


Fig. 3.8. Plots of exact and approximate $\Delta e_d'$.

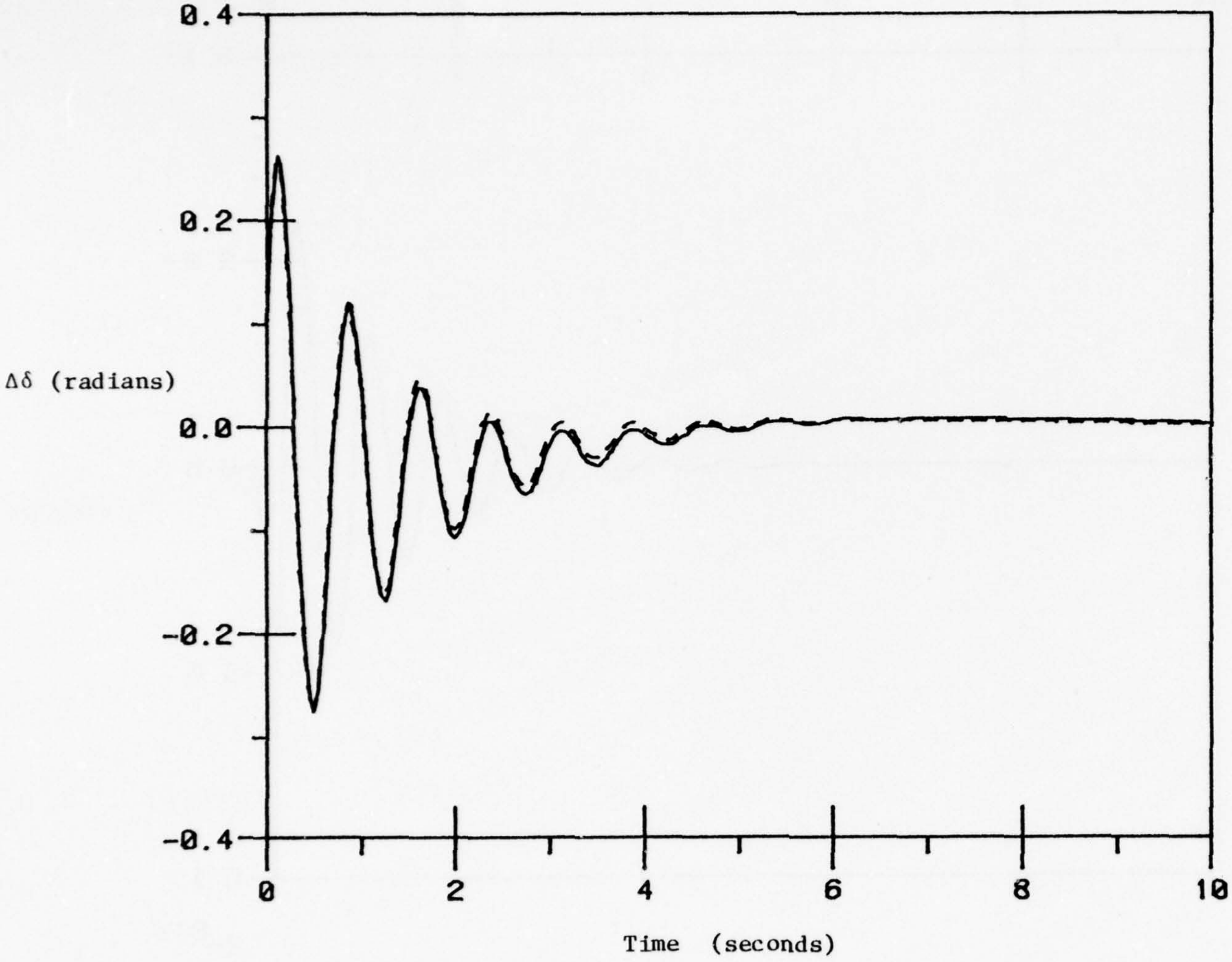


Fig. 3.9. Plots of exact and approximate Δδ.

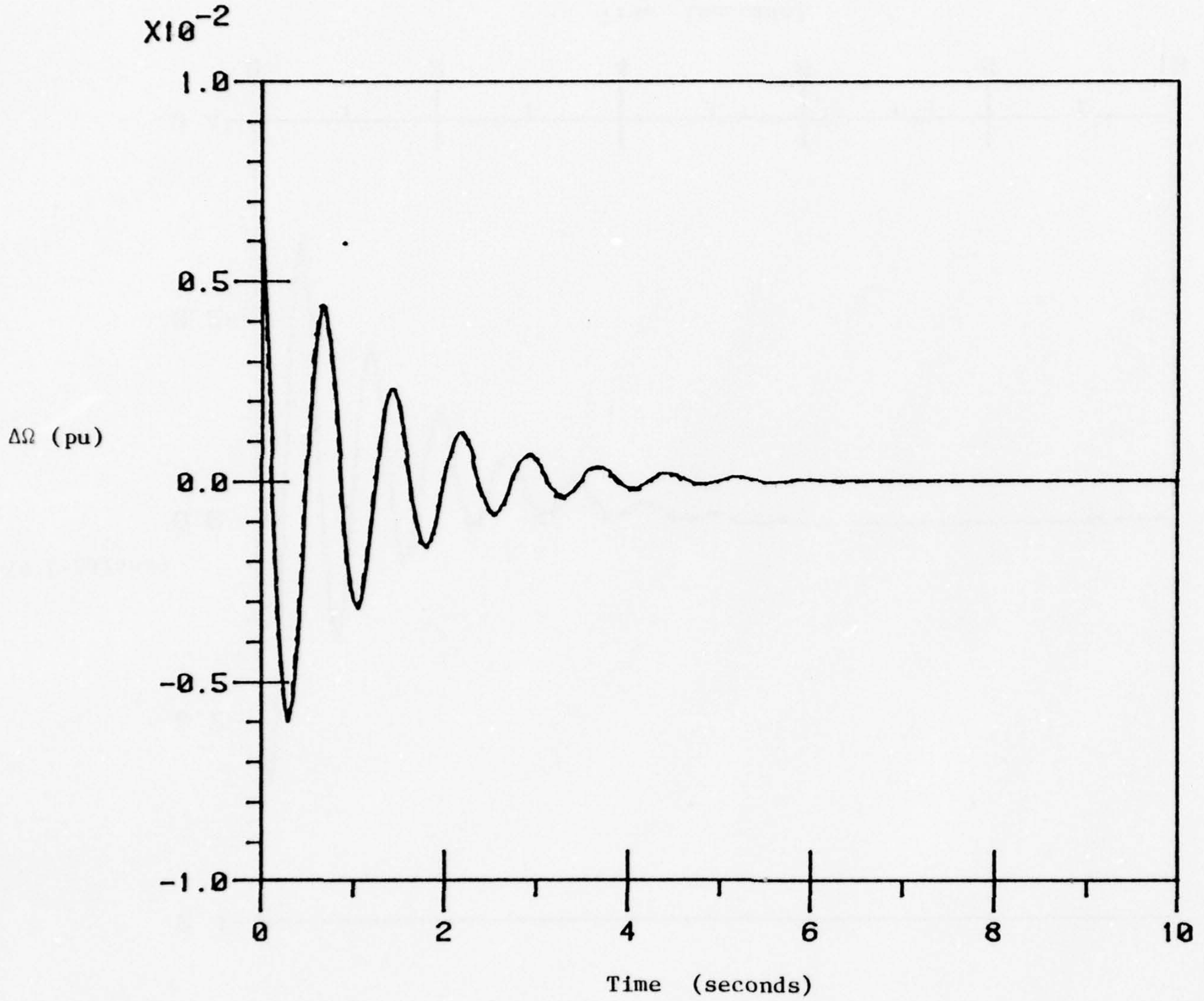


Fig. 3.10. Plots of exact and approximate ΔΩ.

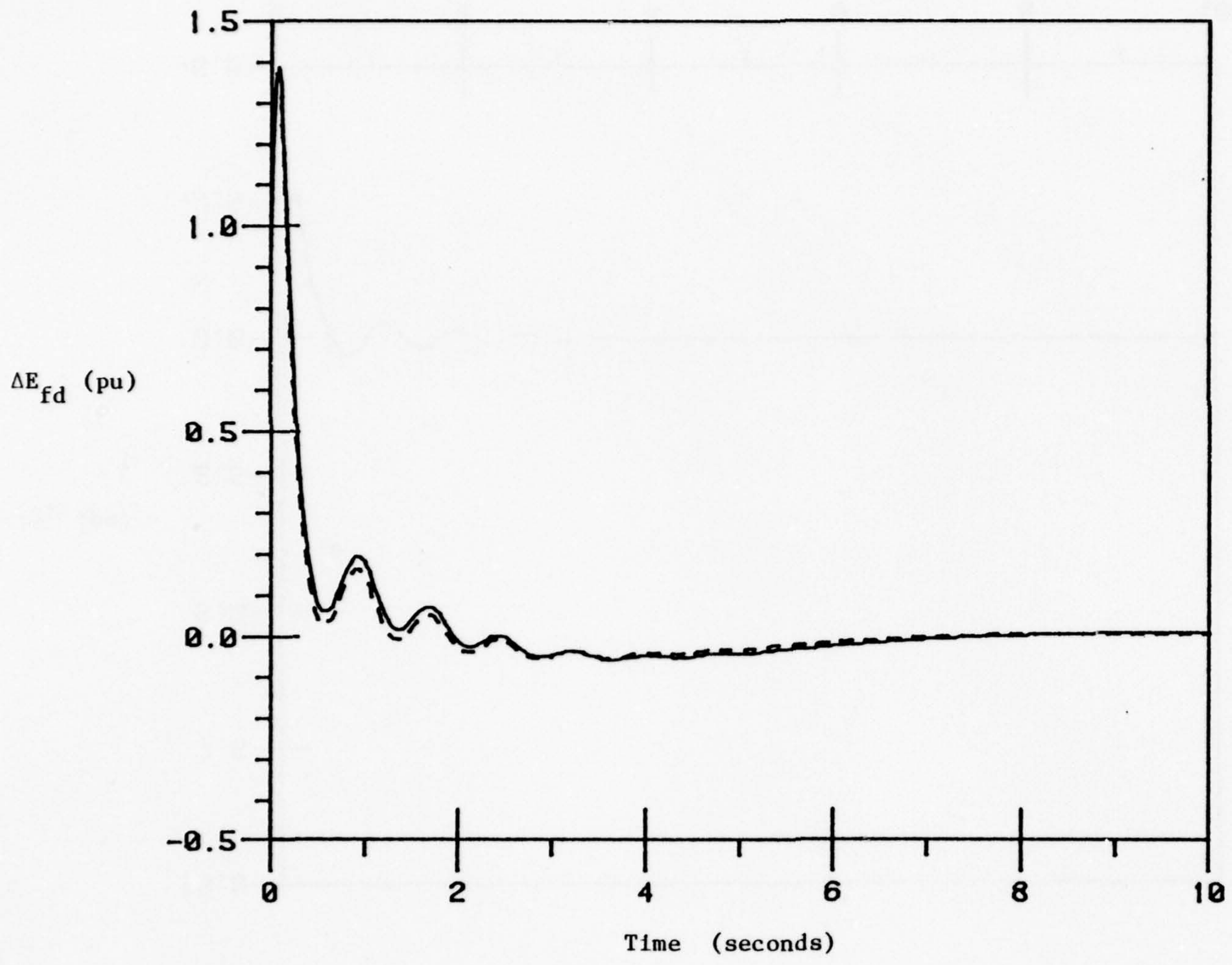


Fig. 3.11. Plots of exact and approximate ΔE_{fd}.

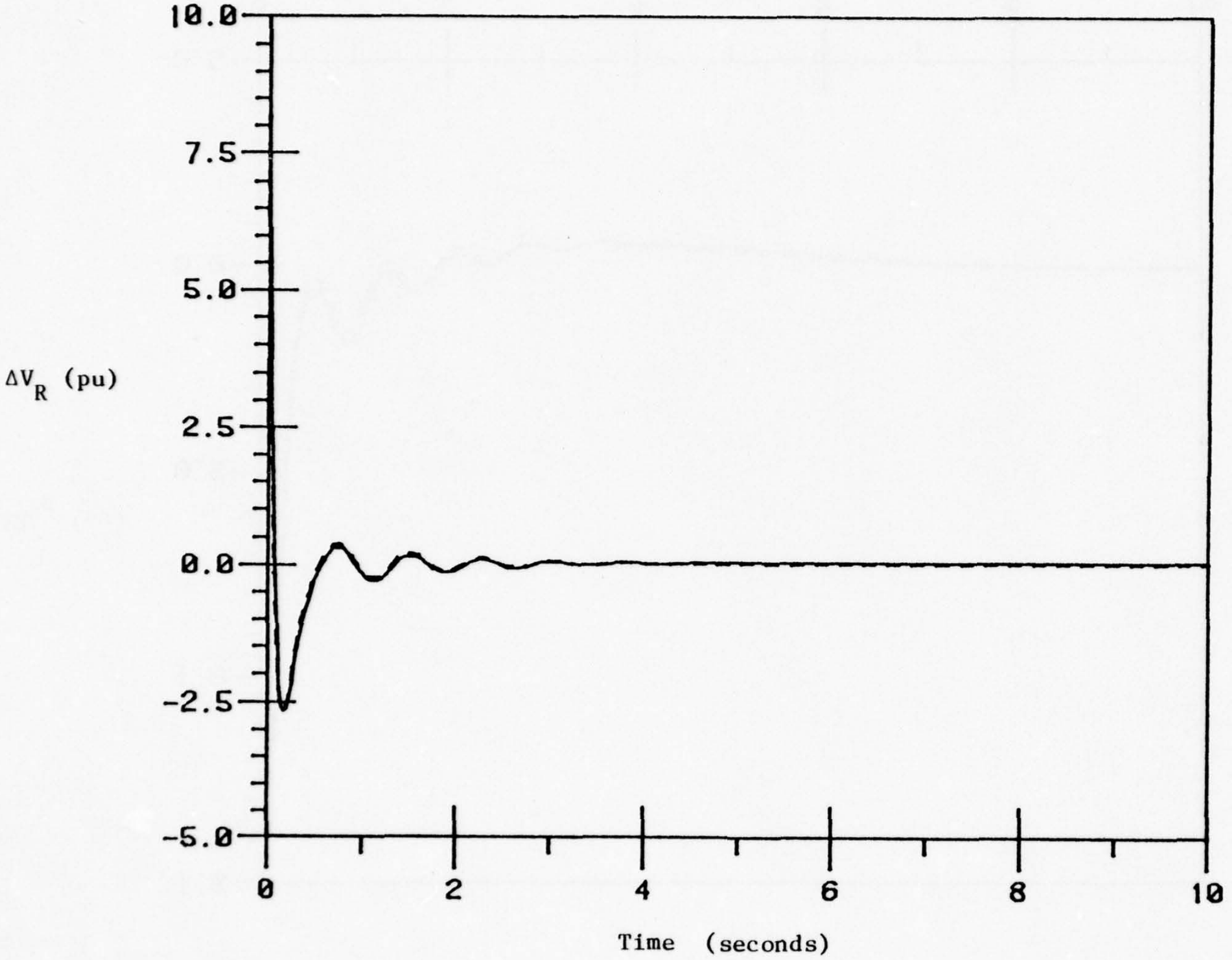


Fig. 3.12. Plots of exact and approximate ΔV_R.

Since there are some discrepancies between the approximate and exact responses using the results of one iteration, the system was simulated again using the results of two iterations. However, in this case all of the approximate responses are indistinguishable from the exact responses.

3.5 Zero-Order Analysis of the Non-Linear System

In this section we examine the usual zero-order approximation (2.2) as applied to the non-linear system (3.3). The calculations involved in the zero-order approximation are quite simple. We have previously identified e'_q and R_f as slow variables and e'_d, δ, Ω, E_{fd}, and V_R as fast variables. We assume that this behavior carries over to the non-linear system. Then, the slow subsystem is

$$\dot{\bar{e}}'_q = \frac{1}{T'_{d_0}}\{ - [1 + (x_d - x')Y]\ \bar{e}'_q - (x_d - x')YV_i \sin\bar{\delta} + \bar{E}_{fd}\} \tag{3.10a}$$

$$\dot{\bar{R}}_f = \frac{1}{T_F}\ (- \bar{R}_f + \frac{K_F}{T_F}\ \bar{E}_{fd}) \tag{3.10b}$$

$$0 = \frac{1}{T'_{q_0}}\{ (x_q - x')YV_i \cos\bar{\delta} - [1 + (x_q - x')Y]\bar{e}'_d\} \tag{3.10c}$$

$$0 = 377\ (\bar{\Omega} - 1) \tag{3.10d}$$

$$0 = \frac{1}{2H}\ [\frac{P_{in}}{\Omega} - YV_i(\bar{e}'_q \cos\bar{\delta} + \bar{e}'_d \sin\bar{\delta}) - D(\bar{\Omega} - 1)] \tag{3.10e}$$

$$0 = \frac{1}{T_E}\{ - [K_E + S_E(\bar{E}_{fd})]\ \bar{E}_{fd} + \bar{V}_R\} \tag{3.10f}$$

$$0 = \frac{1}{T_A}[K_A(\bar{R}_f - \frac{K_F}{T_F}\bar{E}_{fd} - \bar{V}_t + V_{Ref}) - \bar{V}_R] \tag{3.10g}$$

$$\bar{V}_t = [(1-x'Y)^2(\bar{e}'^2_q + \bar{e}'^2_d) + 2(1-x'Y)(x'YV_i)(\bar{e}'_d\cos\bar{\delta} - \bar{e}'_q\sin\bar{\delta}) + (x'YV_i)^2]^{1/2}. \tag{3.10h}$$

Equations (3.10c)-(3.10h) are non-linear, algebraic relationships between the slow variables and the slow part of the fast variables. During numerical integration of the slow subsystem equations, updated values of $\bar{e}'_q$ and $\bar{R}_f$ are used in (3.10c) - (3.10h) to solve for new values of $\bar{e}'_d$, $\bar{\delta}$, $\bar{\Omega}$, $\bar{E}_{fd}$, and $\bar{V}_R$. The updated values of $\bar{\delta}$ and $\bar{E}_{fd}$ are substituted into (3.10a) and (3.10b) to continue the integration for $\bar{e}'_q$ and $\bar{R}_f$.

Once the slow quantities are known, the fast subsystem is integrated. Expressing each fast variable as the sum of its slow and fast parts, the differential equations of the fast subsystem are

$$\dot{\tilde{e}}'_d = \frac{1}{T'_{q_0}}\{(x_q - x')YV_i\cos(\bar{\delta} + \tilde{\delta}) - [1 + (x_q - x')Y](\bar{e}'_d + \tilde{e}'_d)\} \tag{3.11a}$$

$$\dot{\tilde{\delta}} = 377\,\tilde{\Omega} \tag{3.11b}$$

$$\dot{\tilde{\Omega}} = \frac{1}{2H}\{\frac{P_{in}}{\bar{\Omega} + \tilde{\Omega}} - YV_i[\bar{e}'_q\cos(\bar{\delta} + \tilde{\delta}) + (\bar{e}'_d + \tilde{e}'_d)\sin(\bar{\delta} + \tilde{\delta})] - D\tilde{\Omega}\} \tag{3.11c}$$

$$\dot{\tilde{E}}_{fd} = \frac{1}{T_E}\{-[K_E + S_E(\bar{E}_{fd} + \tilde{E}_{fd})](\bar{E}_{fd} + \tilde{E}_{fd}) + \bar{V}_R + \tilde{V}_R\} \tag{3.11d}$$

$$\dot{\tilde{V}}_R = \frac{1}{T_A}\{K_A[\bar{R}_f - \frac{K_F}{T_F}(\bar{E}_{fd} + \tilde{E}_{fd}) - \hat{V}_t + V_{Ref}] - \bar{V}_R - \tilde{V}_R\} \tag{3.11e}$$

$$\hat{V}_t = \{(1-x'Y)^2[\bar{e}'^2_q + (\bar{e}'_d + \tilde{e}'_d)^2] + 2(1-x'Y)(x'YV_i)[(\bar{e}'_d + \tilde{e}'_d)\cos(\bar{\delta} + \tilde{\delta}) - \bar{e}'_q\sin(\bar{\delta} + \tilde{\delta})] + (x'YV_i)^2\}^{1/2}. \tag{3.11f}$$

Since the slow subsystem has already been solved, the slow variables appearing in (3.11) are known during integration of the fast subsystem. The fast subsystem is integrated with a smaller step size than the slow subsystem; hence, in general, it is necessary to interpolate between the known values of the slow variables to obtain the correct values to use in (3.11).

To explore the nature of this zero-order singular perturbation analysis of the non-linear system, we simulate the system response for the same disturbance used in the last section. Figures 3.13-3.19 show the results of this simulation. In each case, the solid curve is the exact response and the dashed curve is the zero-order approximate response. It is quite clear that these responses are very similar to their linear counterparts.

We have previously noted that the slow variables contain a fast component. Since the zero-order approximation of the slow variables does not admit a fast component, e'_q and R_f are not approximated very well by $\bar{e}'_q$ and $\bar{R}_f$. Despite this inaccuracy, the fast variables are approximated quite well. Thus, we are motivated to search for a procedure to correct the slow variable approximation.

3.6 A Correction Method for the Non-Linear System

One method to obtain corrections for non-linear systems is given in [14]. Another, simpler method to approximately account for a fast component in the slow variables will now be given. Consider the system

$$\dot{x} = f(x,z), \quad x(0) = x_0 \; . \tag{3.12}$$

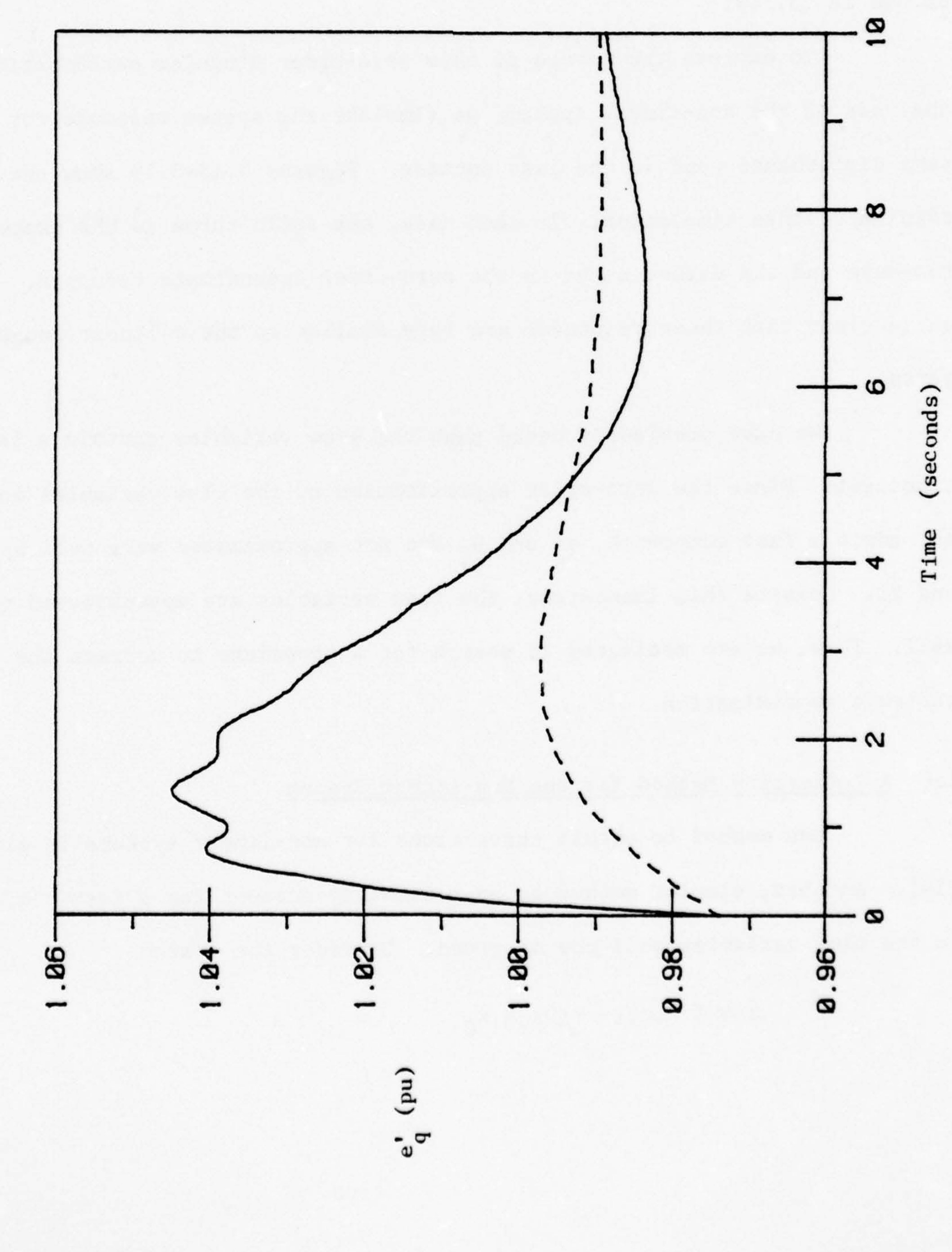


Fig. 3.13. Plots of exact and zero-order approximation of e'_q.

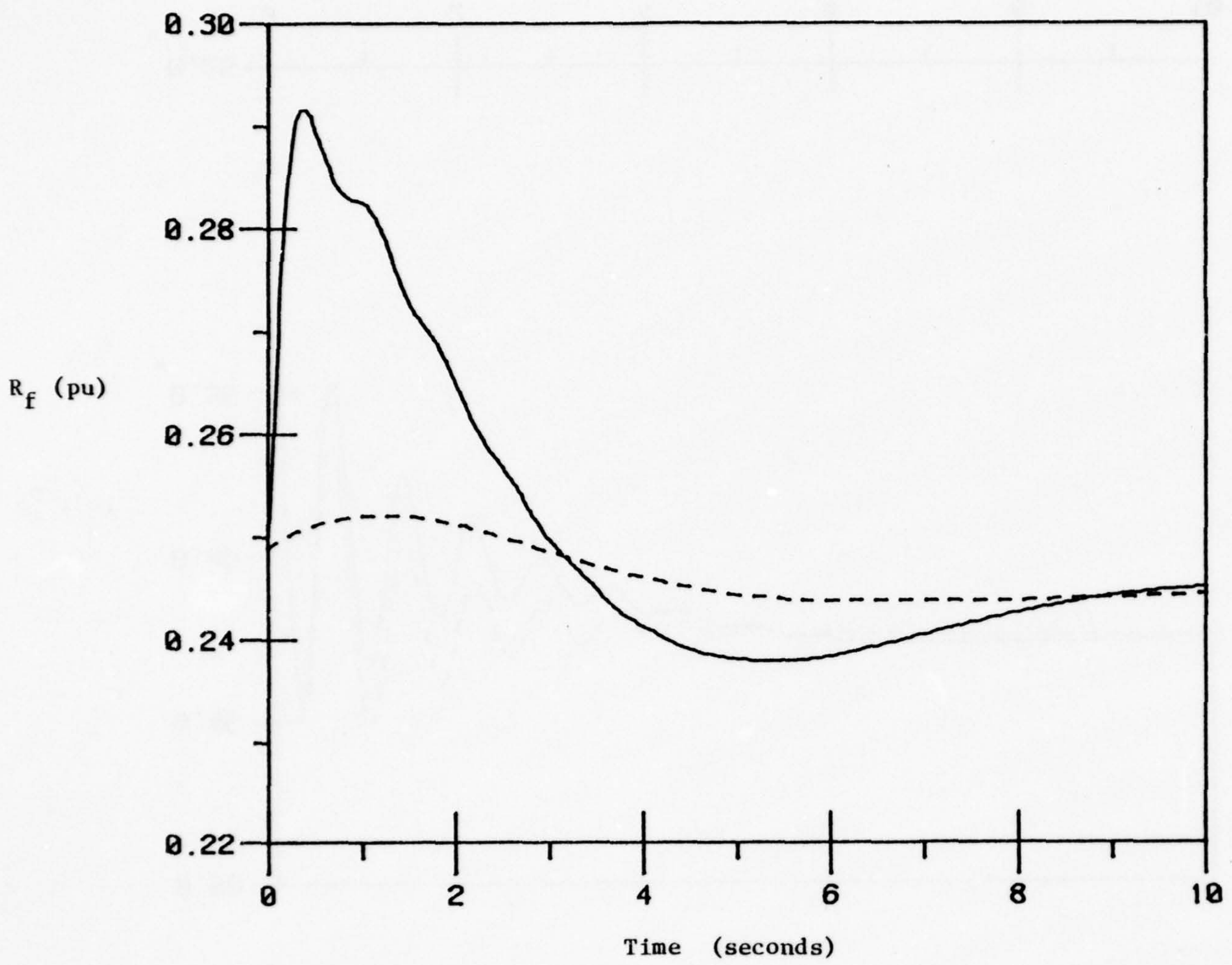


Fig. 3.14. Plots of exact and zero-order approximation of R_f.

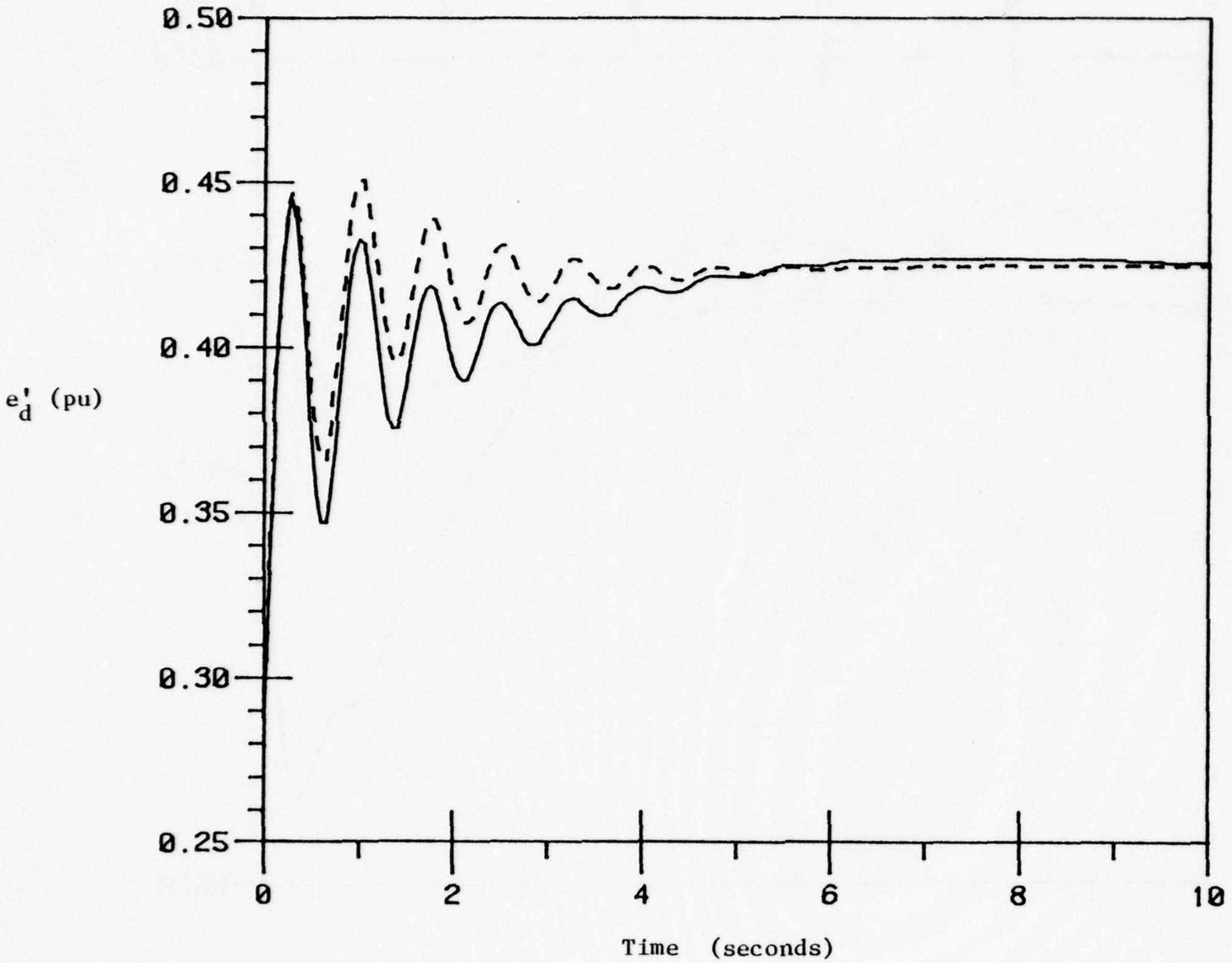


Fig. 3.15. Plots of exact and zero-order approximation of e_d'.

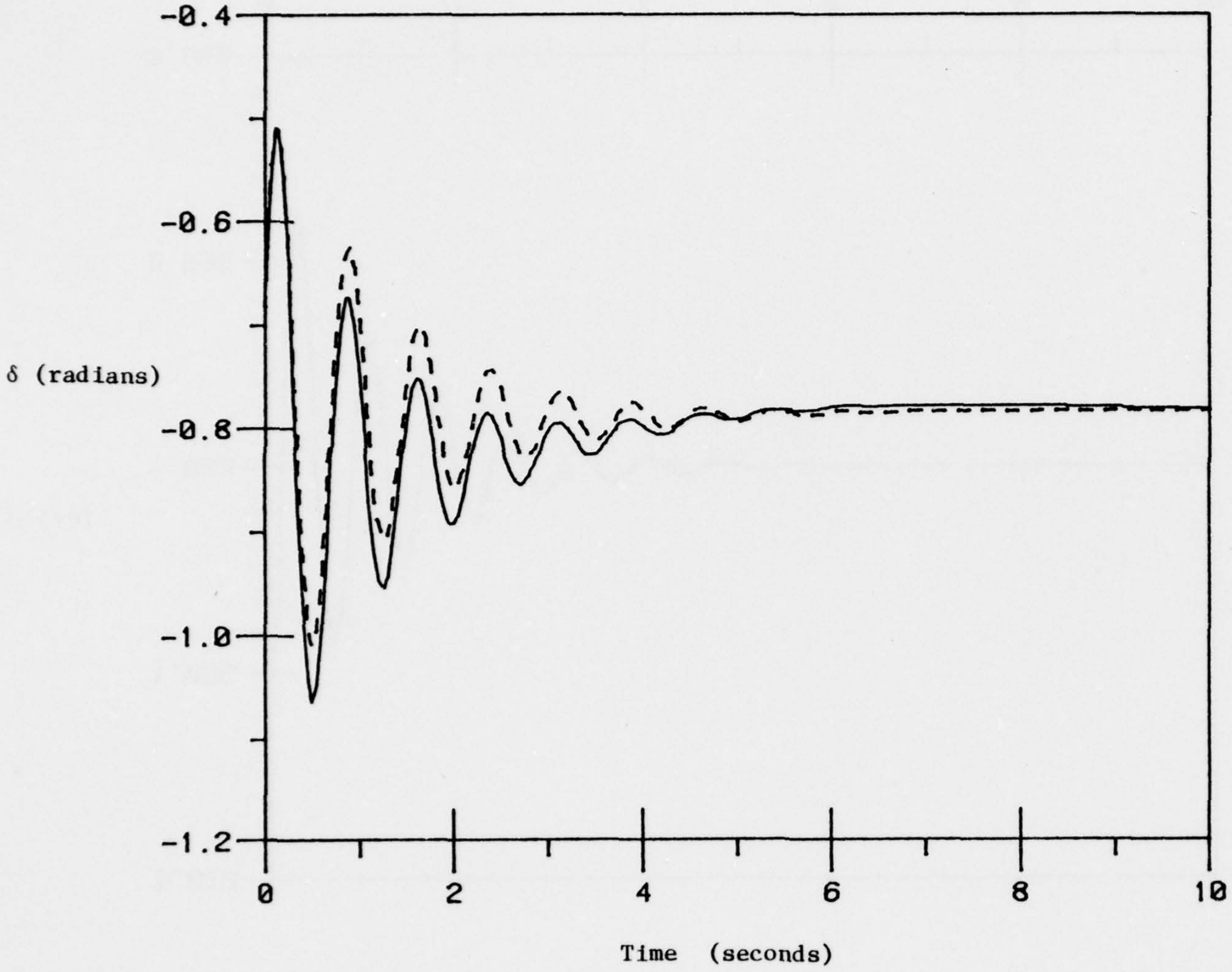


Fig. 3.16. Plots of exact and zero-order approximation of δ.

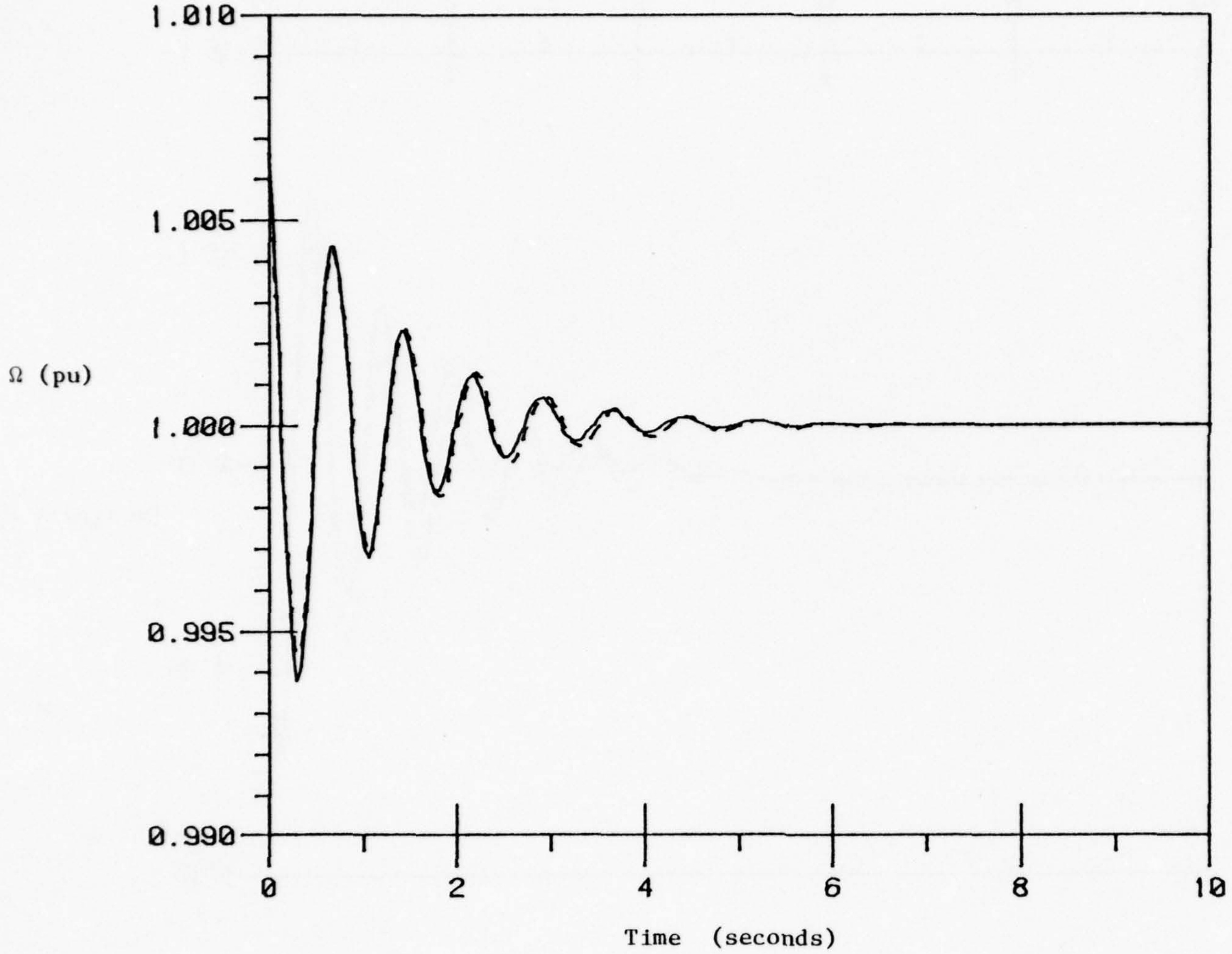


Fig. 3.17. Plots of exact and zero-order approximation of Ω.

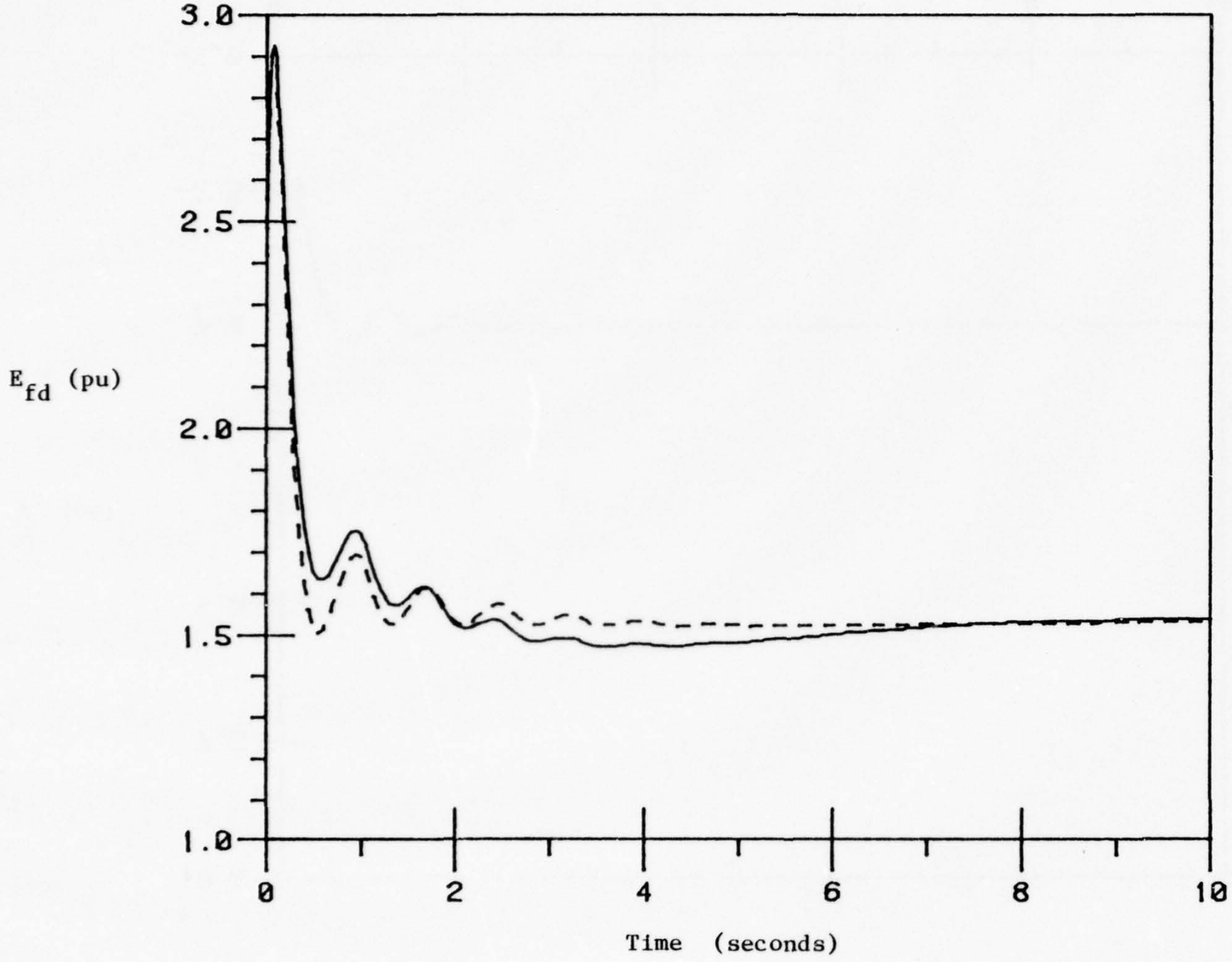


Fig. 3.18. Plots of exact and zero-order approximation of E_{fd}.

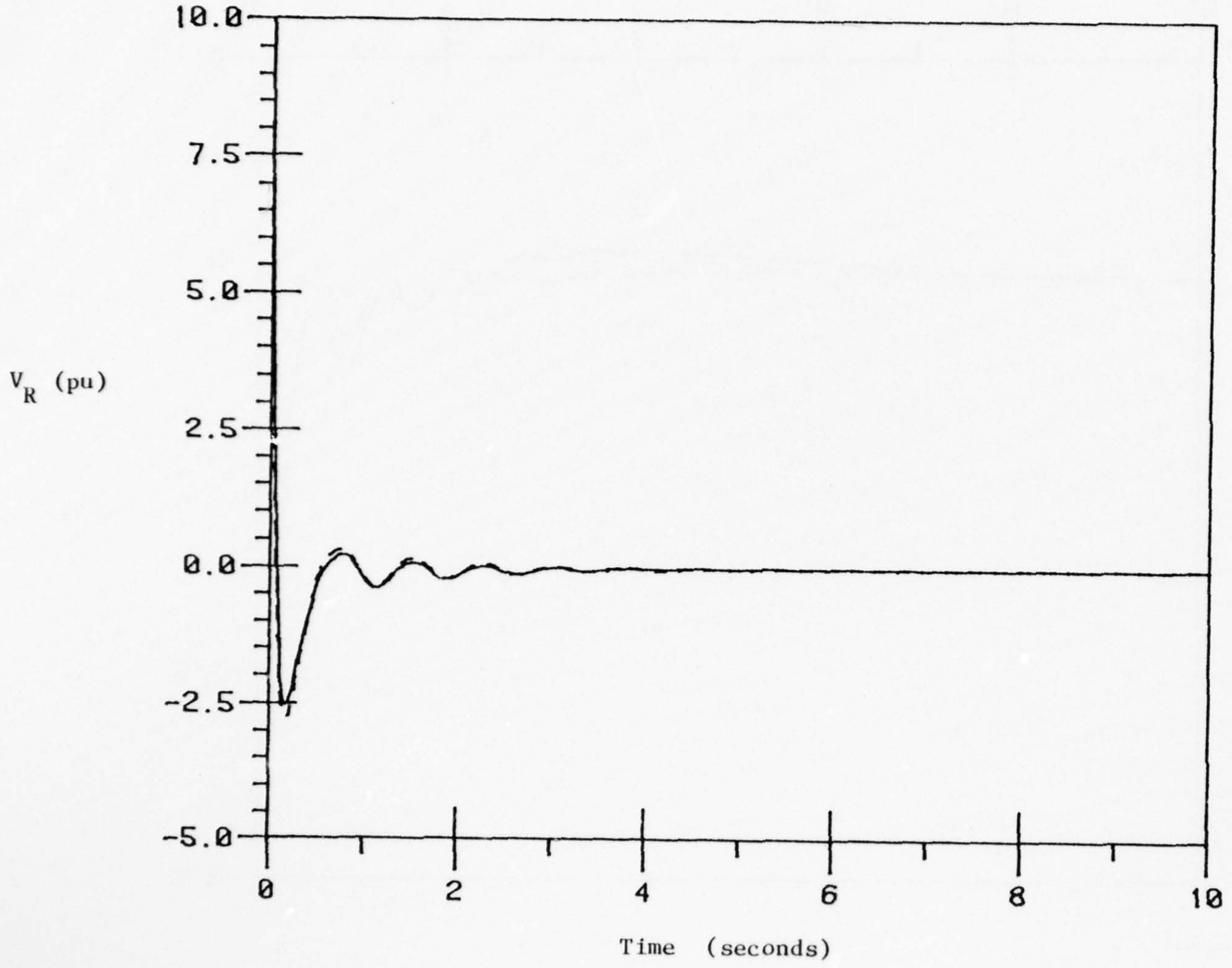


Fig. 3.19. Plots of exact and zero-order approximation of V_R.

Suppose that z is written as

$$z = \varphi(x) + z_f \tag{3.13}$$

where, symbolically, $\varphi(x)$ is the solution of $g(x,z) = 0$ and z_f is the fast part of z. Substituting (3.13) into (3.12) we get

$$\dot{x} = f(x,\varphi(x) + z_f) \ . \tag{3.14}$$

Let f be such that it can be decomposed as follows

$$\begin{aligned} f(x,\varphi(x) + z_f) &= f_1(x,\varphi(x)) + f_2(\varphi(x),\ z_f) \\ &= \hat{f}_1(x) + f_2(\varphi(x),\ z_f) \ . \end{aligned} \tag{3.15}$$

As an approximation, let us use a previously obtained solution for x and z_f in f_2. Let these solutions be denoted by x^o and $z_f^{\ o}$. Then (3.15) becomes

$$\dot{x} \cong \hat{f}_1(x) + f_2(\varphi(x^o), z_f^{\ o}) \ . \tag{3.16}$$

The integral equation form of (3.16) is

$$x - x_0 = \int_0^t \hat{f}_1(x)d\tau + \int_0^t f_2(\varphi(x^o),\ z_f^{\ o})d\tau. \tag{3.17}$$

Let

$$k = \int_0^\infty f_2(\varphi(x^o),\ z_f^{\ o})d\tau. \tag{3.18}$$

We write (3.17) as

$$x = x_o + k + \int_0^t \hat{f}_1(x)d\tau + \int_0^t f_2(\varphi(x^o),\ z_f^{\ o})d\tau - k \ . \tag{3.19}$$

As an approximation, neglect the fast part of x in the first integral. Then the slow part of x is given by

$$\dot{x}_s \cong \hat{f}_1(x_s), \quad x_s(0) = x_0 + k \ , \tag{3.20}$$

and the fast part of x is approximated by

$$x_f \cong -k + \int_0^t f_2(\varphi(x^o), z_f^{\,o})\, d\tau \quad . \tag{3.21}$$

Notice in (3.20) how the presence of fast variables in the slow equations can change the slow initial conditions.

Once the slow subsystem has been solved with the corrected initial conditions, the fast subsystem can be resolved using the corrected values of x, namely

$$\mu\dot{z}_f = g(x_s + x_f, \ z_f + z_s), \ z_f(0) = z_0 - z_s(0) \tag{3.22a}$$

$$0 = g(x_s, z_s) \ . \tag{3.22b}$$

Note that this scheme could be employed iteratively.

Applying this method first to (3.3a) we find that the slow part of e'_q is obtained as the solution of

$$\dot{e}'_{q_s} = \frac{1}{T'_{d_0}}\{ - [1 + (x_d - x')Y]\ e'_{q_s} - (x_d - x')YV_i \sin\delta_s + E_{fd_s}\},$$
$$e'_{q_s}(0) = e'_q(0) + k_1 \tag{3.23}$$

and the fast part of e'_q is

$$e'_{q_f} = -k_1 - \frac{(x_d - x')YV_i}{T'_{d_0}} \int_0^t [\sin\delta_s^{\,o}(-1 + \cos\delta_f^{\,o}) +$$
$$\cos\delta_s^{\,o}\sin\delta_f^{\,o}]d\tau + \frac{1}{T'_{d_0}}\int_0^t E^o_{fd_f}\, d\tau \tag{3.24}$$

and

$$k_1 = -\frac{(x_d - x')YV_i}{T'_{d_0}} \int_0^\infty [\sin\delta_s{}^o(-1 + \cos\delta_f{}^o) +$$

$$\cos\delta_s{}^o \sin\delta_f{}^o]d\tau + \frac{1}{T'_{d_0}} \int_0^\infty E^o_{fd_f} d\tau. \tag{3.25}$$

Similarly for (3.3b), the slow part of R_f is solved from

$$\dot{R}_{f_s} = \frac{1}{T_F}(-R_{f_s} + \frac{K_F}{T_F} E_{fd_s}),\ R_{f_s}(0) = R_f(0) + k_2 \tag{3.26}$$

and the fast part is

$$R_{f_f} = -k_2 + \frac{K_F}{T_F{}^2} \int_0^t E^o_{fd_f}\, d\tau \tag{3.27}$$

where

$$k_2 = \frac{K_F}{T_F{}^2} \int_0^\infty E^o_{fd_f}\, d\tau\ . \tag{3.28}$$

δ_s and E_{fd_s} appearing in (3.23) and (3.26) are solved from the same algebraic equations as in the original approximation. The integrals in (3.24), (3.25), (3.27), and (3.28) can be evaluated by any convenient numerical integration scheme since the arguments appearing under the integral sign are known functions of time. We also note that it is not necessary to carry out the integrals in (3.25) and (3.28) to infinite upper limits. From Fig. 3.16 and 3.18 we observe that δ^o_f and $E^o_{fd_f}$ both become approximately zero for t > 5 seconds.

Finally the fast subsystem is solved as in (3.11) but this time, both the slow and fast parts of e'_q and R_f are substituted instead of just the slow part.

To test this procedure, we use the same system disturbance as in previous sections. In carrying out the indicated calculations, the zero-order approximation is obtained first. The zero-order fast solution is used to evaluate the necessary integrals. Using the corrected initial conditions, the slow subsystem is solved again. To complete the procedure, the corrected responses $e'_{q_s} + e'_{q_f}$ and $R_{f_s} + R_{f_f}$ are substituted into the fast subsystem equations and this subsystem is solved again.

The results of the simulation are shown in Figs. 3.20-3.26. In general the agreement between the exact and approximate responses is good. Regarding the slow variables, it is clear that the correction procedure has not only accounted for the fast component but also corrected the initial conditions. Using these corrected slow solutions during integration of the fast subsystem yields generally excellent fast approximations. Hence, in this case, the correction procedure for non-linear systems performs adequately.

3.7 Conclusions

In this chapter we have verified that the single machine-infinite bus system possesses the separation of time scales property. Using the growth of model and block diagram techniques, we have discovered which dynamics of the model are primarily responsible for each mode. It was found that e'_d, δ, Ω, E_{fd}, and V_R constitute a suitable fast subsystem while e'_q and R_f comprise the slow subsystem.

We have applied linear singular perturbation methods to the linearized seventh order model. In particular, one iteration of the iterative technique produces quite acceptable eigenvalue and time response approximations. Two iterations give almost perfect approximations. In the

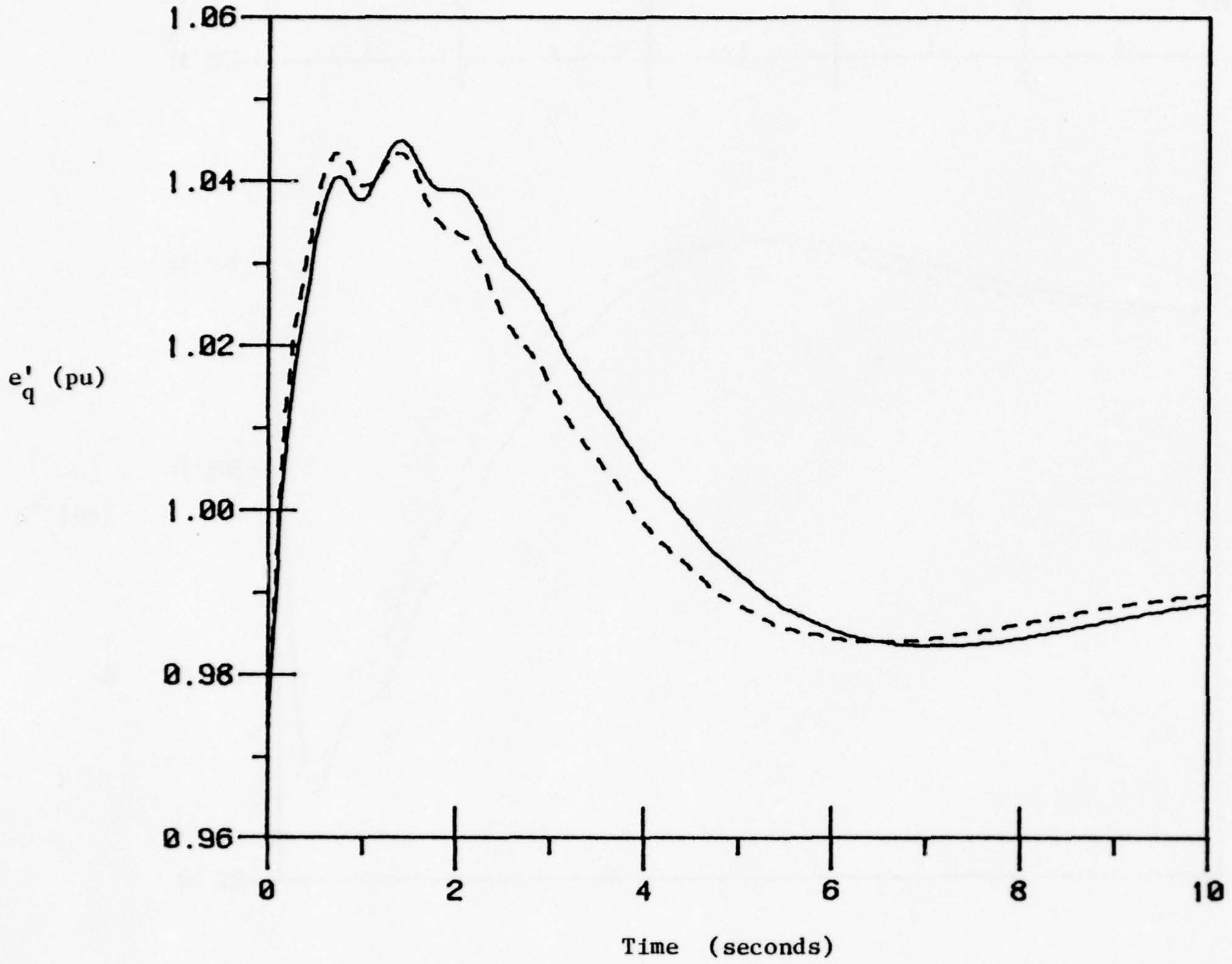


Fig. 3.20. Plots of exact and corrected approximation of e'_q.

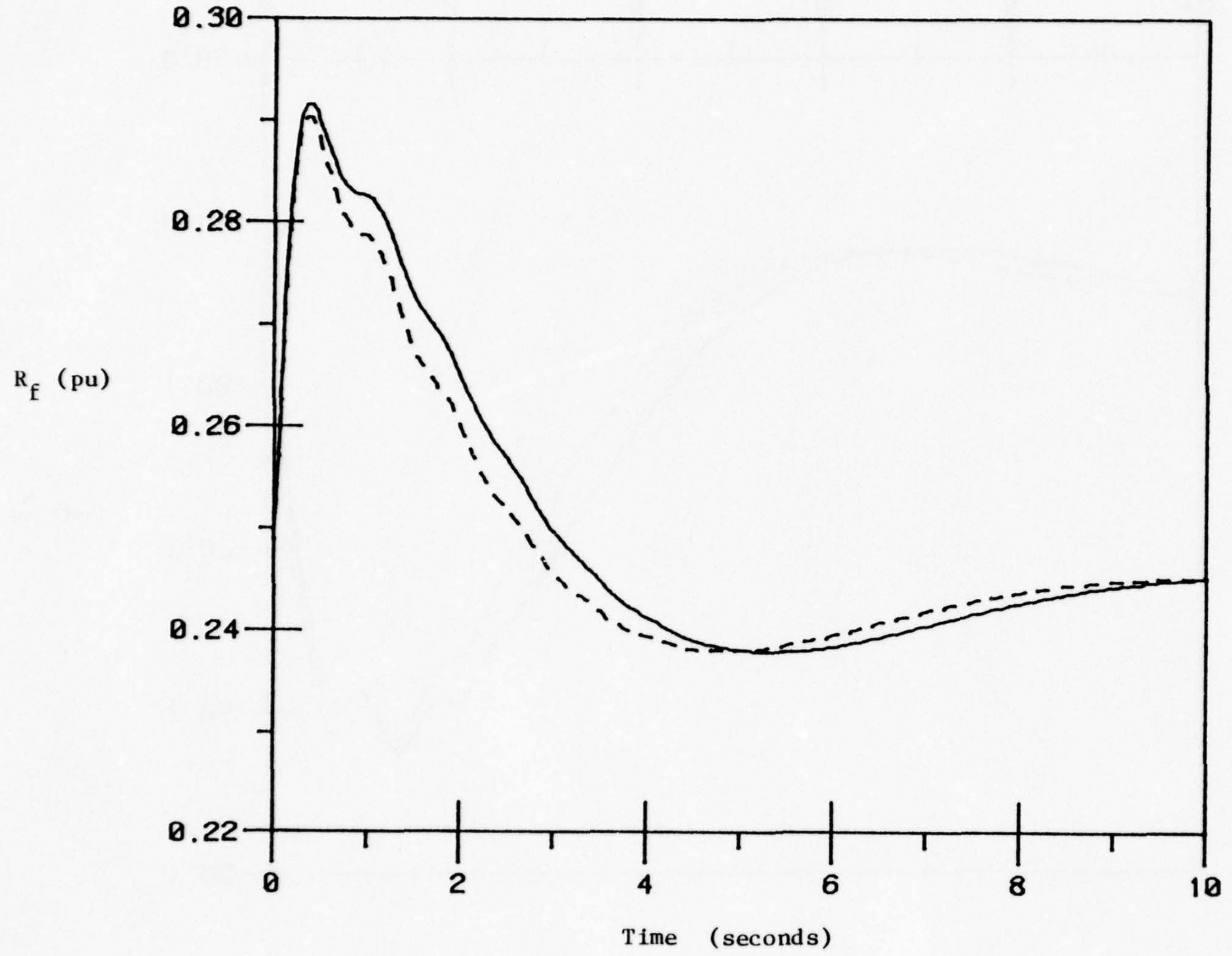


Fig. 3.21. Plots of exact and corrected approximation of R_f.

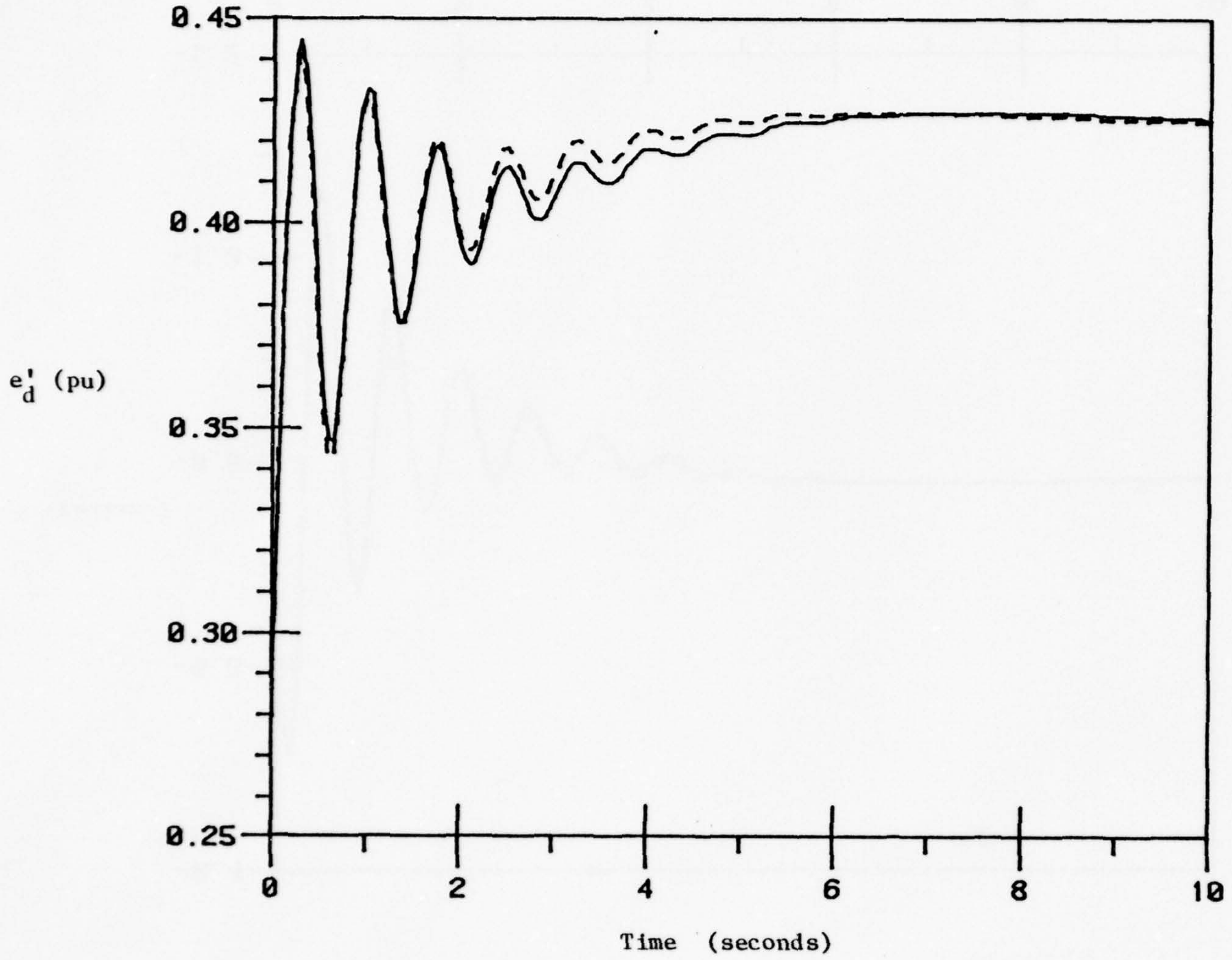


Fig. 3.22. Plots of exact and corrected approximation of e'_d.

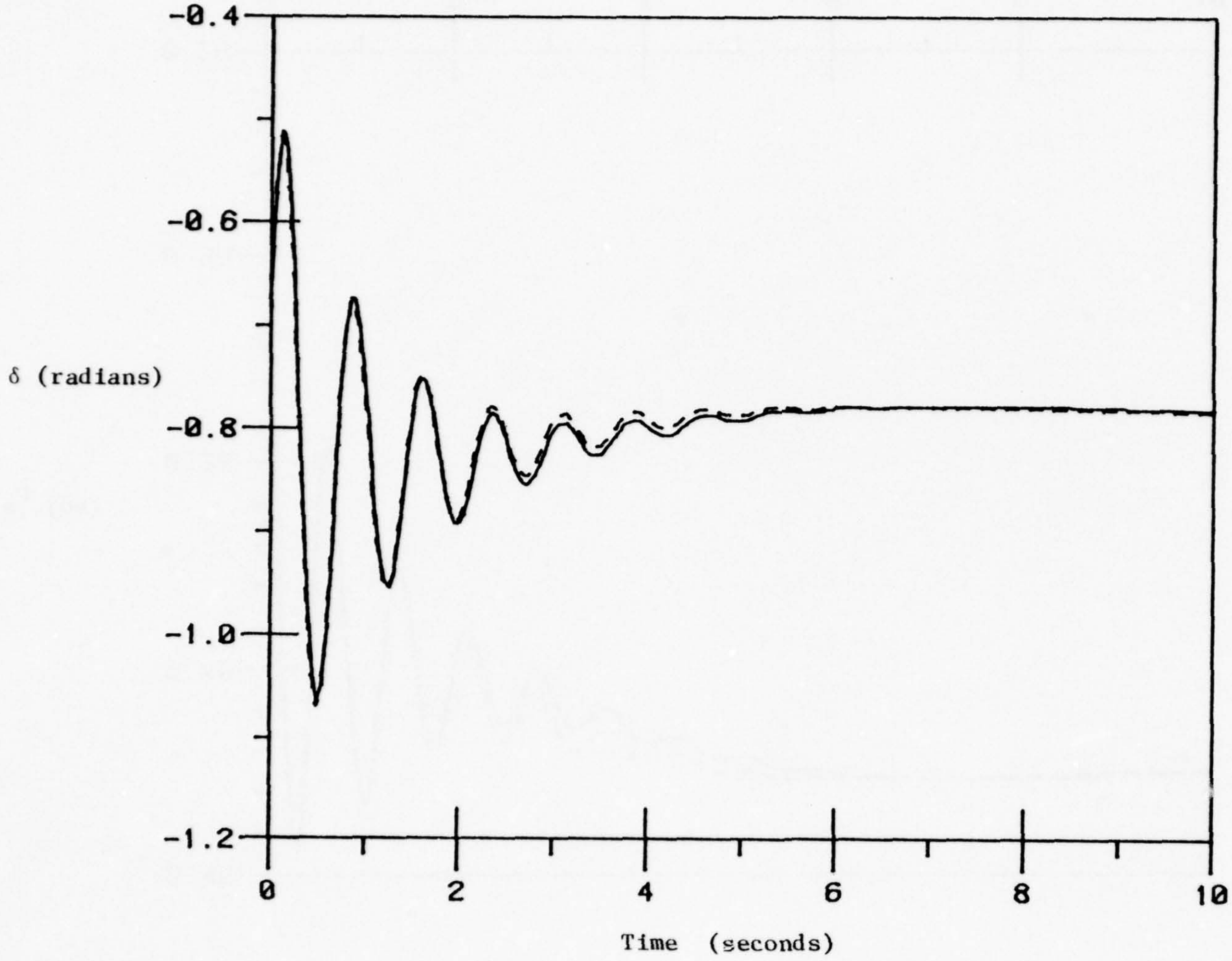


Fig. 3.23. Plots of exact and corrected approximation of δ.

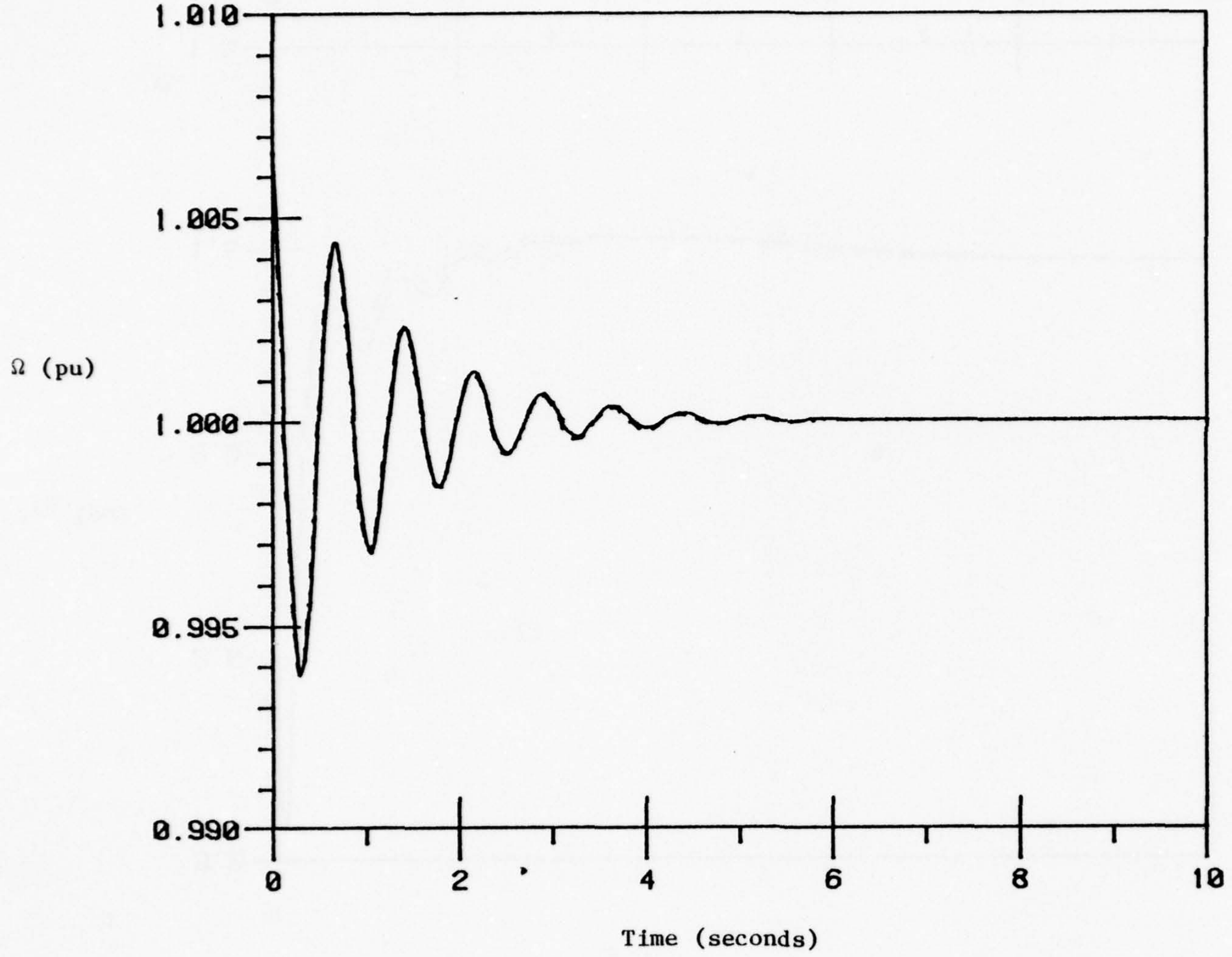


Fig. 3.24. Plots of exact and corrected approximation of Ω.

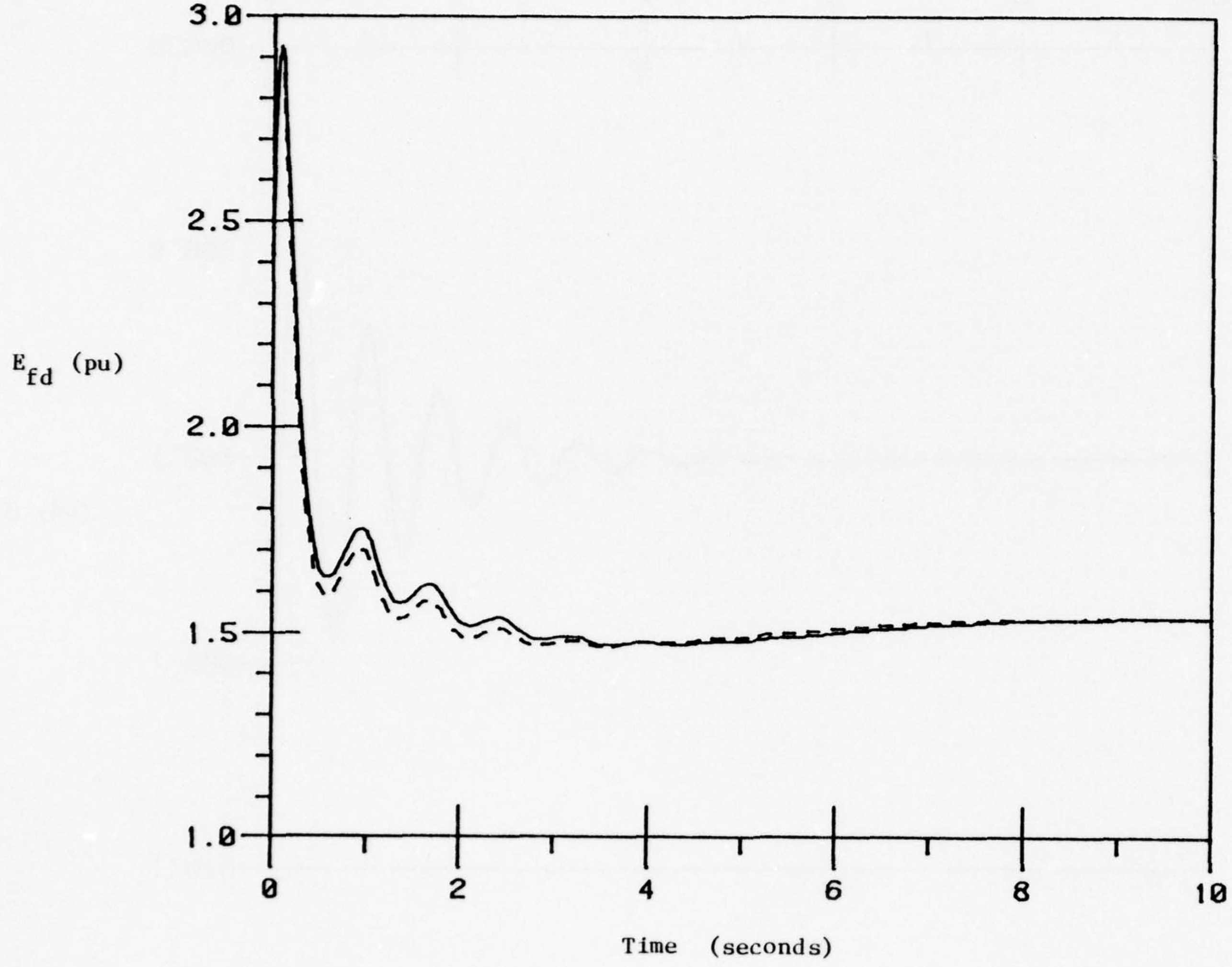


Fig. 3.25. Plots of exact and corrected approximation of E_{fd}.

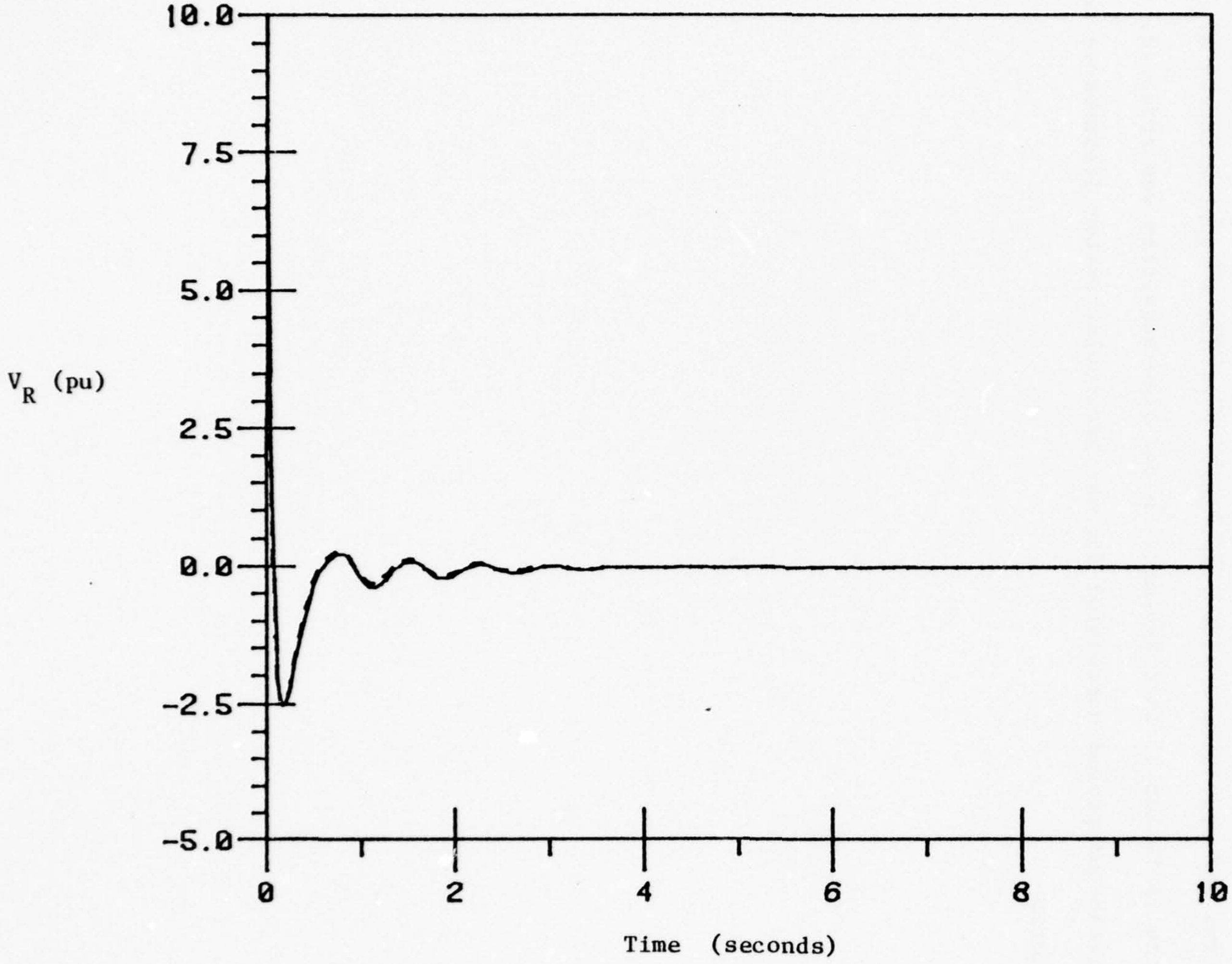


Fig. 3.26. Plots of exact and corrected approximation of V_R.

simulations, we noted that the slow variables can have a significant fast component. The iterative scheme for linear systems automatically accounts for fast components in the slow variables.

Finally, we investigated the behavior of the zero-order singular perturbation approximation for the non-linear model. We found that the forcing effect of the field voltage can cause discrepancies between the true and approximate slow responses. A method to approximately account for the inclusion of fast dynamics in the slow subsystem was proposed. This technique proved successful for the particular system disturbance considered here.

4. STUDY OF MULTI-MACHINE SYSTEM

4.1. Introduction

In this chapter we generalize the results of Chapter 3 by studying a multi-machine system. We find this system to be very similar to the single machine system. In particular, the variables which we previously identified as slow and fast remain in these respective subsystems. But we also find an additional slow mode due to the system frequency drift. The properties of the angle-speed modes are clarified when the original angles and speeds are transformed to some new angles and speeds.

We present both linear and non-linear results. The linear iteration technique does not encounter any difficulties with this system. However, for the particular disturbance considered, the slow variables generally contain a significant fast component. Thus, the zero-order non-linear approximation suffers from the same difficulty as in the single machine case. The method presented in Chapter 3 to correct the slow approximation is found to improve the approximations relative to the zero-order results.

4.2. System Description

The system studied in this chapter is the 9 bus, 3 machine system shown in Fig. 4.1. The transmission line data are given in Table 2 and the synchronous machine parameters are given in Table 3. The exciter parameters are the same for all three machines and are identical to the settings used in Chapter 3. The disturbance used throughout this chapter is a 3 phase, 5 cycle fault applied near bus 8 which is cleared by opening the line from bus 8 to bus 9.

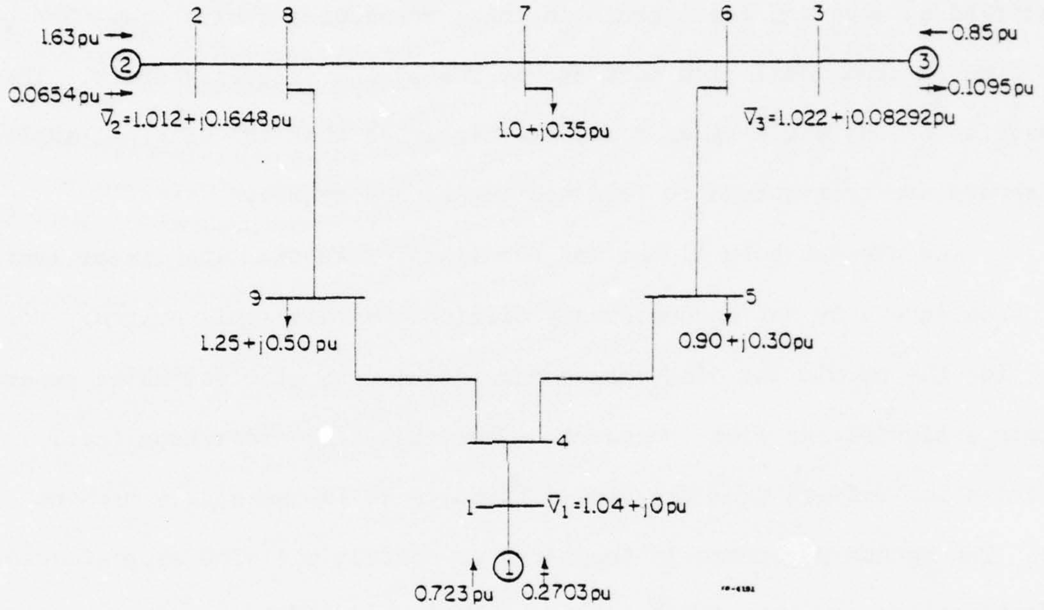


Fig. 4.1. 3 machine, 9 bus test system.
Note: All load flow information is for predisturbance conditions.
The base power is 100 MVA.

Table 2. Transmission System Data
(100 MVA Base)

Line #	From	To	R(pu)	X(pu)	B/2 (pu)
1	1	4	0	0.0567	0
2	4	5	0.017	0.092	0.079
3	5	6	0.039	0.170	0.179
4	3	6	0	0.0586	0
5	6	7	0.0119	0.1008	0.1045
6	7	8	0.0085	0.072	0.0745
7	8	2	0	0.0625	0
8	8	9	0.032	0.161	0.153
9	9	4	0.01	0.085	0.088

Table 3. Synchronous Machine Data
(100 MVA Base)

Parameter	Machine #		
	1	2	3
x_d(pu)	0.6	0.8958	0.9
x_q(pu)	0.58	0.8645	0.85
x_ℓ(pu)	0.056	0.110	0.18
x'_d(pu)	0.0608	0.1198	0.1813
x'_q(pu)	0.0608	0.1198	0.1813
T'_{d_0} (sec)	4.0	6.0	5.0
T'_{q_0} (sec)	0.25	0.54	0.65
r_a(pu)	0	0	0
H(sec)	23.64	6.4	3.01
D(pu)	9.6	2.5	1.0

We now discuss the choice of angle and speed variables for describing this system. The well known non-linear equations for the swing model are

$$\dot{\delta}_{i1} = 377(\Omega_i - \Omega_1), \qquad i = 2,3 \tag{4.1a}$$

$$\dot{\Omega}_i = \frac{1}{2H_i}[P_{in_i} - D_i(\Omega_i - 1) - \sum_{j=1}^{3} V_i' V_j' Y_{ij} \cos(\theta_{ij} + \delta_{j1} - \delta_{i1})] , \qquad i = 1,2,3 . \tag{4.1b}$$

When (4.1) is linearized about the equilibrium point corresponding to the post-disturbance network, the resulting system matrix is found to be

$$\begin{bmatrix} 0.0 & 0.0 & -377.0 & 377.0 & 0.0 \\ 0.0 & 0.0 & -377.0 & 0.0 & 377.0 \\ 0.0144 & 0.0235 & -0.203 & 0.0 & 0.0 \\ -0.131 & 0.0930 & 0.0 & -0.195 & 0.0 \\ 0.216 & -0.371 & 0.0 & 0.0 & -0.161 \end{bmatrix} .$$

The eigenvalues of this matrix are

$$-0.0857 \pm j12.90$$

$$-0.0978 \pm j6.09$$

$$-0.197 .$$

The mode -0.197 is the well known frequency drift mode which corresponds to the common movement of all of the machine speeds to the new equilibrium speed. The two complex pairs are rotor oscillation modes. Since one "frequency" is more than twice the other "frequency", we suspect that these modes may be attributed to something other than individual rotor oscillations. In particular, with the removal of the line from bus 8 to bus 9, machine 2 is relatively weakly connected to machine 1. However, the coupling between machines 2 and 3 remains relatively strong. Thus, it is likely that one of the oscillatory modes represents the common oscillation of machines 2 and 3 with respect to machine 1; while the other mode is the oscillation of machine 2 with respect to machine 3. This situation is similar to the one described in [6].

To more clearly exhibit the behavior described above, we introduce the following change of variables. First, since the system frequency drift involves all of the rotating inertia of the system, we define the system frequency to be

$$\Omega_r = \frac{H_1\Omega_1 + H_2\Omega_2 + H_3\Omega_3}{H} \qquad (4.2a)$$

where

$$H = H_1 + H_2 + H_3 \ . \qquad (4.2b)$$

Second, to display the combined motion of machines 2 and 3, we define δ_c and Ω_c as

$$\delta_c = \frac{H_1}{hH} (H_2\delta_{21} + H_3\delta_{31}) \qquad (4.3a)$$

$$\Omega_c = \frac{H_2\Omega_2 + H_3\Omega_3}{h} - \Omega_r \qquad (4.3b)$$

where

$$h = H_2 + H_3 \ . \qquad (4.3b)$$

Finally, the oscillation of machine 2 with respect to machine 3 is described by

$$\delta_d = \delta_{21} - \delta_{31} \qquad (4.4a)$$

$$\Omega_d = \Omega_2 - \Omega_3 \ . \qquad (4.4b)$$

These variables are similar to the ones used by Stanton in [15].

In these variables, the system matrix is

$$\begin{array}{c} \Delta\Omega_r \\ \Delta\delta_c \\ \Delta\Omega_c \\ \Delta\delta_d \\ \Delta\Omega_d \end{array} \left[\begin{array}{c|cc|cc} -0.198 & 0.00776 & 0.00486 & 0.000733 & -0.00181 \\ \hline 0.0 & 0.0 & 377.0 & 0.0 & 0.0 \\ 0.0122 & -0.113 & -0.191 & 0.0304 & -0.00454 \\ \hline 0.0 & 0.0 & 0.0 & 0.0 & 377.0 \\ -0.0292 & 0.164 & -0.0292 & -0.426 & -0.175 \end{array}\right].$$

The eigenvalues of the indicated diagonal blocks are, respectively -0.198, -0.0954 ± j6.54, and -0.0877 ± j12.68. These values correspond quite favorably with the eigenvalues of the original matrix indicating that this choice of variables is more appropriate for describing the system than the original set of angle-speed variables.

Using the new angle-speed variables along with the basic equations derived in Appendix A, the complete set of equations for this system is

$$\dot{\Omega}_r = \frac{1}{2H}\Bigg[\frac{P_{in_1}}{\Omega_r - \frac{h}{H_1}\Omega_c} + \frac{P_{in_2}}{\Omega_r + \Omega_c + \frac{H_3}{h}\Omega_d} + \frac{P_{in_3}}{\Omega_r + \Omega_c - \frac{H_2}{h}\Omega_d}$$

$$- D(\Omega_r - 1) + (D_1\frac{h}{H_1} - D_2 - D_3)\Omega_c + (-D_2\frac{H_3}{h} + D_3\frac{H_2}{h})\Omega_d$$

$$- \sum_{i=1}^{3}(e'_{q_i} i_{q_i} + e'_{d_i} i_{d_i})\Bigg] \qquad (4.5a)$$

$$\dot{R}_{f_i} = \frac{1}{T_{F_i}}(-R_{f_i} + \frac{K_{F_i}}{T_{F_i}} E_{fd_i}), \qquad i = 1,2,3 \qquad (4.5b)$$

$$\dot{e}'_{q_i} = \frac{1}{T'_{d_{0i}}}[-e'_{q_i} - (x_{d_i} - x'_i)i_{d_i} + E_{fd_i}], \qquad i = 1,2,3 \qquad (4.5c)$$

$$\dot{e}'_{d_i} = \frac{1}{T'_{q_{0i}}} [-e'_{d_i} + (x_{q_i} - x'_i) i_{q_i}] \quad , \qquad i = 1,2,3 \tag{4.5d}$$

$$\dot{V}_{R_i} = \frac{1}{T_{A_i}} [K_{A_i} (R_{f_i} - \frac{K_{F_i}}{T_{F_i}} E_{fd_i} - V_{t_i} + V_{Ref_i}) - V_{R_i}], \quad i = 1,2,3 \tag{4.5e}$$

$$\dot{E}_{fd_i} = \frac{1}{T_{E_i}} \{ - [K_{E_i} + S_E(E_{fd_i})] E_{fd_i} + V_{R_i} \} \quad , \qquad i = 1,2,3 \tag{4.5f}$$

$$\delta_c = 377\, \Omega_c \tag{4.5g}$$

$$\dot{\Omega}_c = \frac{1}{2H} \{ (\frac{H_1}{h}) [\frac{P_{in_2}}{\Omega_r + \Omega_c + \frac{H_3}{h} \Omega_d} + \frac{P_{in_3}}{\Omega_r + \Omega_c - \frac{H_2}{h} \Omega_d}] - \frac{P_{in_1}}{\Omega_r - \frac{h}{H_1} \Omega_c}$$

$$+ [D_1 - (\frac{H_1}{h})(D_2 + D_3)](\Omega_r - 1) - [D_1 \frac{h}{H_1} + (\frac{H_1}{h})(D_2 + D_3)]\, \Omega_c$$

$$+ [(\frac{H_1}{h^2})(-H_3 D_2 + H_2 D_3)] \Omega_d + e'_{q_1} i_{q_1} + e'_{d_1} i_{d_1}$$

$$- (\frac{H_1}{h}) \sum_{i=2}^{3} (e'_{q_i} i_{q_i} + e'_{d_i} i_{d_i}) \} \tag{4.5h}$$

$$\dot{\delta}_d = 377\, \Omega_d \tag{4.5i}$$

$$\dot{\Omega}_d = \frac{1}{2} \{ \frac{P_{in_2}/H_2}{\Omega_r + \Omega_c + \frac{H_3}{h} \Omega_d} - \frac{P_{in_3}/H_3}{\Omega_r + \Omega_c - \frac{H_2}{h} \Omega_d} + (-\frac{D_2}{H_2} + \frac{D_3}{H_3})(\Omega_r - 1)$$

$$+ (-\frac{D_2}{H_2} + \frac{D_3}{H_3}) \Omega_c - \frac{1}{h} (\frac{H_3}{H_2} D_2 + \frac{H_2}{H_3} D_3) \Omega_d$$

$$- \frac{1}{H_2} (e'_{q_2} i_{q_2} + e'_{d_2} i_{d_2}) + \frac{1}{H_3} (e'_{q_3} i_{q_3} + e'_{d_3} i_{d_3}) \} \tag{4.5j}$$

where

$$D = D_1 + D_2 + D_3 \tag{4.5k}$$

$$V_{t_i} = \sqrt{v_{d_i}^2 + v_{q_i}^2} \tag{4.5l}$$

$$v_{d_i} = e'_{d_i} + x'_i \, i_{q_i} \tag{4.5m}$$

$$v_{q_i} = e'_{q_i} - x'_i \, i_{d_i} \quad . \tag{4.5n}$$

The currents appearing in (4.5) are

$$\begin{aligned} i_{d_1} = {} & Y_{11}(e'_{d_1} \cos \theta_{11} - e'_{q_1} \sin \theta_{11}) \\ & + Y_{12}[e'_{d_2} \cos(\theta_{12} + \frac{H}{H_1} \delta_c + \frac{H_3}{h} \delta_d) - e'_{q_2} \sin(\theta_{12} + \frac{H}{H_1} \delta_c + \frac{H_3}{h} \delta_d)] \\ & + Y_{13}[e'_{d_3} \cos(\theta_{13} + \frac{H}{H_1} \delta_c - \frac{H_2}{h} \delta_d) - e'_{q_3} \sin(\theta_{13} + \frac{H}{H_1} \delta_c - \frac{H_2}{h} \delta_d)] \end{aligned} \tag{4.6a}$$

$$\begin{aligned} i_{q_1} = {} & Y_{11}(e'_{q_1} \cos \theta_{11} + e'_{d_1} \sin \theta_{11}) \\ & + Y_{12}[e'_{q_2} \cos(\theta_{12} + \frac{H}{H_1} \delta_c + \frac{H_3}{h} \delta_d) + e'_{d_2} \sin(\theta_{12} + \frac{H}{H_1} \delta_c + \frac{H_3}{h} \delta_d)] \\ & + Y_{13}[e'_{q_3} \cos(\theta_{13} + \frac{H}{H_1} \delta_c - \frac{H_2}{h} \delta_d) + e'_{d_3} \sin(\theta_{13} + \frac{H}{H_1} \delta_c - \frac{H_2}{h} \delta_d)] \end{aligned} \tag{4.6b}$$

$$\begin{aligned} i_{d_2} = {} & Y_{22}(e'_{d_2} \cos \theta_{22} - e'_{q_2} \sin \theta_{22}) \\ & + Y_{21}[e'_{d_1} \cos(\theta_{21} - \frac{H}{H_1} \delta_c - \frac{H_3}{h} \delta_d) - e'_{q_1} \sin(\theta_{21} - \frac{H}{H_1} \delta_c - \frac{H_3}{h} \delta_d)] \\ & + Y_{32}[e'_{d_3} \cos(\theta_{32} - \delta_d) - e'_{q_3} \sin(\theta_{32} - \delta_d)] \end{aligned} \tag{4.6c}$$

$$\begin{aligned} i_{q_2} &= Y_{22}(e'_{q_2}\cos\theta_{22} + e'_{d_2}\sin\theta_{22}) \\ &+ Y_{21}[e'_{q_1}\cos(\theta_{21} - \frac{H}{H_1}\delta_c - \frac{H_3}{h}\delta_d) + e'_{d_1}\sin(\theta_{21} - \frac{H}{H_1}\delta_c - \frac{H_3}{h}\delta_d)] \\ &+ Y_{23}[e'_{q_3}\cos(\theta_{23} - \delta_d) + e'_{d_3}\sin(\theta_{23} - \delta_d)] \end{aligned} \tag{4.6d}$$

$$\begin{aligned} i_{d_3} &= Y_{33}(e'_{d_3}\cos\theta_{33} - e'_{q_3}\sin\theta_{33}) \\ &+ Y_{31}[e'_{d_1}\cos(\theta_{31} - \frac{H}{H_1}\delta_c + \frac{H_2}{h}\delta_d) - e'_{q_1}\sin(\theta_{31} - \frac{H}{H_1}\delta_c + \frac{H_2}{h}\delta_d)] \\ &+ Y_{32}[e'_{d_2}\cos(\theta_{32} + \delta_d) - e'_{q_2}\sin(\theta_{32} + \delta_d)] \end{aligned} \tag{4.6e}$$

$$\begin{aligned} i_{q_3} &= Y_{33}(e'_{q_3}\cos\theta_{33} + e'_{d_3}\sin\theta_{33}) \\ &+ Y_{31}[e'_{q_1}\cos(\theta_{31} - \frac{H}{H_1}\delta_c + \frac{H_2}{h}\delta_d) + e'_{d_1}\sin(\theta_{31} - \frac{H}{H_1}\delta_c + \frac{H_2}{h}\delta_d)] \\ &+ Y_{32}[e'_{q_2}\cos(\theta_{32} + \delta_d) + e'_{d_2}\sin(\theta_{32} + \delta_d)] \quad . \end{aligned} \tag{4.6f}$$

4.3 Determination of Partitioning

In the previous section, using the swing model, we found the location of the swing and drift modes. The next step in the growth of model method for determining the system partitioning is to examine the eigenvalues of the flux decay model. Unfortunately, the flux decay model does not have an equilibrium for the post-disturbance network chosen for this study. Thus, the linearized equations cannot be obtained. However, the complete model (4.5) does possess an equilibrium and the eigenvalues of the linearized equations fall into groups which are reminiscent of the location of eigenvalues in the single machine case. Hence, we can use our experience with the single machine system to determine which dynamics are primarily responsible for each mode. Having made this determination, we can proceed to partition the system into slow and fast subsystems.

The system matrix for the complete model is shown in Fig. 4.2. The eigenvalues of this matrix are

$$
\begin{array}{l}
-1.26 \pm j12.95 \\
-8.16 \pm j7.69 \\
-8.56 \pm j8.26 \\
-8.50 \pm j8.09 \\
-0.988 \pm j5.35 \\
-6.90 \\
-4.45 \\
-1.90 \\
-1.22 \pm j0.989 \\
-0.245 \\
-0.505 \pm j0.761 \\
-0.289 \pm j0.480 \ .
\end{array}
$$

Based on our examination of the swing model in the last section, it is clear that the pair -1.26 ± j12.95 is primarily due to $\Delta\delta_c$ and $\Delta\Omega_c$ while the pair -0.988 ± j5.35 arises from $\Delta\delta_d$ and $\Delta\Omega_d$. The real root -0.245 is the system frequency drift mode and, hence, is mainly determined by $\Delta\Omega_r$.

Now, using the experience gained during our study of the single machine system, it is clear that the pairs -8.16 ± j7.69, -8.56 ± j8.26, and -8.50 ± j8.09 are mainly produced by the interaction of ΔE_{fd} and ΔV_R in the three individual voltage regulators. Furthermore, the pairs -1.22 ± j0.989, -0.505 ± j0.761, and -0.289 ± j0.480 obviously arise from the combination of $\Delta e_q'$ and ΔR_f in the respective machine models. Finally, the three $\Delta e_d''$s are primarily responsible for the three real roots, -6.90, -4.45, and -1.90.

$$
\begin{array}{c|cccccccccccccccccccc}
\Delta\Omega_r & -0.246 + & + & + & + & -0.0459 + & -0.0140 + & -0.0180 + & -0.00482 + & -0.0166 + & -0.00890 + & + & + & + & + & + & + & 0.00744 + & -0.0278 + & 0.000850 + & -0.000962 \\
\Delta R_{f_1} & + & -1.0 + & + & + & + & + & + & + & + & + & + & 0.16 + & + & + & + & + & + & + & + & \\
\Delta R_{f_2} & + & + & -1.0 + & + & + & + & + & + & + & + & + & + & + & 0.16 + & + & + & + & + & + & \\
\Delta R_{f_3} & + & + & + & -1.0 + & + & + & + & + & + & + & + & + & + & + & + & 0.16 + & + & + & + & \\
\Delta e'_{q_1} & + & + & + & + & -0.560 + & 0.0561 + & 0.131 + & -0.154 + & 0.0789 + & 0.0634 + & + & 0.25 + & + & + & + & + & -0.0464 + & + & -0.00583 + & \\
\Delta e'_{q_2} & + & + & + & + & 0.0236 + & -0.427 + & 0.124 + & -0.0898 + & -0.0484 + & -0.0982 + & + & + & + & 0.167 + & + & + & -0.140 + & + & -0.180 + & \\
\Delta e'_{q_3} & + & + & + & + & 0.109 + & 0.165 + & -0.538 + & -0.110 + & 0.0610 + & -0.0387 + & + & + & + & + & + & 0.20 + & -0.198 + & + & 0.155 + & \\
\Delta e'_{d_1} & + & + & + & + & 2.36 + & -1.22 + & -0.977 + & -8.77 + & 0.864 + & 2.02 + & + & + & + & + & + & + & -5.43 + & + & 1.12 + & \\
\Delta e'_{d_2} & + & + & + & + & 0.958 + & 0.516 + & 1.05 + & 0.251 + & -4.63 + & 1.32 + & + & + & + & + & + & + & 0.0677 + & + & 0.835 + & \\
\Delta e'_{d_3} & + & + & + & + & 0.789 + & -0.437 + & 0.277 + & 0.781 + & 1.18 + & -3.96 + & + & + & + & + & + & + & 0.905 + & + & -1.71 + & \\
\Delta V_{R_1} & + & 417 + & + & + & -355 + & -6.60 + & -21.0 + & -59.5 + & -17.0 + & -17.6 + & -16.7 + & -66.7 + & + & + & + & + & 24.6 + & + & -2.26 + & \\
\Delta E_{fd_1} & + & + & + & + & + & + & + & + & + & + & 2.0 + & 0.0838 + & + & + & + & + & + & + & + & \\
\Delta V_{R_2} & + & + & 417 + & + & -32.0 + & -223 + & -60.0 + & 16.0 + & -225 + & -10.9 + & + & + & -16.7 + & -66.7 + & + & + & 33.9 + & + & 23.0 + & \\
\Delta E_{fd_2} & + & + & + & + & + & + & + & + & + & + & + & + & 2.0 + & 0.0829 + & + & + & + & + & + & \\
\Delta V_{R_3} & + & + & + & 417 + & -79.4 + & -56.2 + & -213 + & 18.6 + & -73.2 + & -110 + & + & + & + & + & -16.7 + & -66.7 + & 52.7 + & + & -2.11 + & \\
\Delta E_{fd_3} & + & + & + & + & + & + & + & + & + & + & + & + & + & + & 2.0 + & 0.0843 + & + & + & + & \\
\Delta\delta_c & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & 377 + & + & \\
\Delta\Omega_c & -0.0699 + & + & + & + & -0.000252 + & -0.0732 + & -0.0833 + & -0.0840 + & -0.0253 + & 0.0266 + & + & + & + & + & + & + & -0.124 + & -0.288 + & 0.0329 + & -0.00242 \\
\Delta\delta_d & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & + & 377 \\
\Delta\Omega_d & -0.0155 + & + & + & + & 0.0327 + & -0.348 + & 0.318 + & 0.128 + & 0.163 + & -0.373 + & + & + & + & + & + & + & 0.179 + & -0.0155 + & -0.454 + & -0.310
\end{array}
$$

Fig. 4.2. System matrix for full model.
Note: only non-zero entries shown.

It is apparent that this system behaves, in many ways, like three single machine systems. Thus, for partitioning the system into fast and slow subsystems, the variables should be grouped per phenomenon (as opposed to per machine) using the previously identified fast and slow variables as a basis for the grouping. Of course we must include the new variable, $\Delta\Omega_r$, which does not appear in the single machine system.

Based on this discussion and the system eigenvalues, we choose as the slow subsystem the variables $\Delta\Omega_r$, ΔR_{f_1}, ΔR_{f_2}, ΔR_{f_3}, $\Delta e'_{q_1}$, $\Delta e'_{q_2}$, and $\Delta e'_{q_3}$. The remaining variables constitute the fast subsystem.

4.4 Linear System Study

The eigenvalue approximations of the full system along with their exact counterparts are shown in Table 4. We notice that most eigenvalues are approximated very closely after the second iteration. However, the approximations corresponding to exact modes at -1.22 ± j0.989 and -1.90 are not completely satisfactory after only two iterations. In fact, it takes six iterations to obtain reasonable approximations for these modes.

Regarding this latter behavior, we observe that -1.22 ± j0.989 is the largest of the slow eigenvalues while -1.90 is the smallest of the fast eigenvalues. The ratio of their magnitudes is 1.21; hence, there is no clear separation between them. Based on this fact, we can speculate that these eigenvalues are somewhat sensitive and therefore more difficult to approximate than the others. In particular, these eigenvalues converge to their correct values non-uniformly.

The next eigenvalue to the left is -4.45. The separation between -4.45 and -1.90 is more favorable than the separation between -1.90 and -1.22 ± j0.989. Hence we are led to believe that if the variable which is

Table 4. Exact and Approximate Eigenvalues for 20th Order System

Subsystem	Eigenvalues						
	Approximate - Iteration #						Exact
	1	2	3	4	5	6	
Slow	-1.10 ± j0.684	-1.26 ± j0.816	-1.33 ± j0.911	-1.34 ± j1.00	-1.30 ± j1.07	-1.22 ± j1.09	-1.22 ± j0.989
	-0.245	-0.245	-0.245	-0.245	-0.245	-0.245	-0.245
	-0.546 ± j0.753	-0.500 ± j0.760	-0.503 ± j0.759	-0.505 ± j0.759	-0.506 ± j0.760	-0.506 ± j0.761	-0.505 ± j0.761
	-0.312 ± j0.513	-0.285 ± j0.482	-0.289 ± j0.480	-0.289 ± j0.480	-0.289 ± j0.480	-0.289 ± j0.480	-0.289 ± j0.480
Fast	-1.26 ± j12.9	-1.26 ± j13.0	-1.26 ± j13.0	-1.26 ± j13.0	-1.26 ± j13.0	-1.26 ± j13.0	-1.26 ± j13.0
	-1.01 ± j5.39	-0.996 ± j5.34	-0.988 ± j5.35	-0.988 ± j5.35	-0.988 ± j5.35	-0.988 ± j5.35	-0.988 ± j5.35
	-2.04	-1.81	-1.68	-1.65	-1.74	-1.89	-1.90
	-6.95	-6.91	-6.90	-6.90	-6.90	-6.90	-6.90
	-4.50	-4.46	-4.45	-4.45	-4.45	-4.45	-4.45
	-8.16 ± j7.70	-8.16 ± j7.70	-8.16 ± j7.69	-8.16 ± j7.69	-8.16 ± j7.69	-8.16 ± j7.69	-8.16 ± j7.69
	-8.45 ± j8.09	-8.50 ± j8.10	-8.50 ± j8.09	-8.50 ± j8.09	-8.50 ± j8.09	-8.50 ± j8.09	-8.50 ± j8.09
	-8.53 ± j8.27	-8.56 ± j8.26	-8.56 ± j8.26	-8.56 ± j8.26	-8.56 ± j8.26	-8.56 ± j8.26	-8.56 ± j8.26

responsible for the -1.90 mode were placed in the slow subsystem, this non-uniformity of convergence might not occur. By examining the $\Delta e'_d$ block in the system matrix, we see that the eigenvalue -1.90 is primarily determined by $\Delta e'_{d_3}$. So a more reasonable choice for the slow subsystem would include $\Delta e'_{d_3}$ with the variables already selected.

The simulations reveal some interesting characteristics of this system. The time responses for the post-disturbance network are given in Figs. 4.3 - 4.22. The initial conditions for these linear simulations were calculated by the same procedure as was used to calculate the linear initial conditions in Chapter 3. In each case, the solid curve is the exact solution while the dotted curve is the approximation obtained using the results of three iterations of the iterative procedure. It was found that if four or more iterations were performed, the exact and approximate curves were virtually indistinguishable.

We note that, as in the single machine case, the slow variables (Figs. 4.3-4.9) contain fast components. In addition, it appears that the machine 1 slow variables contain more of the rotor angle oscillations (particularly the lower frequency oscillations) than their machine 2 and 3 counterparts. This observation suggests that the mode $-1.22 \pm j0.989$ is a machine 1 mode. Thus, $\Delta e'_{q_1}$ and ΔR_{f_1} respond more strongly to the fast phenomena than the machine 2 and 3 slow variables. The effect of large field voltage (ΔE_{fd}) is readily observed in the $\Delta e'_q$'s and ΔR_f's as was the case in the single machine system. The slow part of all of the slow variables is clearly present.

The response of $\Delta\Omega_r$ (Fig. 4.3) is very interesting. $\Delta\dot{\Omega}_r$ is proportional to the mismatch between total system real power generation and total load. Approximately the first 4 seconds of the $\Delta\Omega_r$ response is

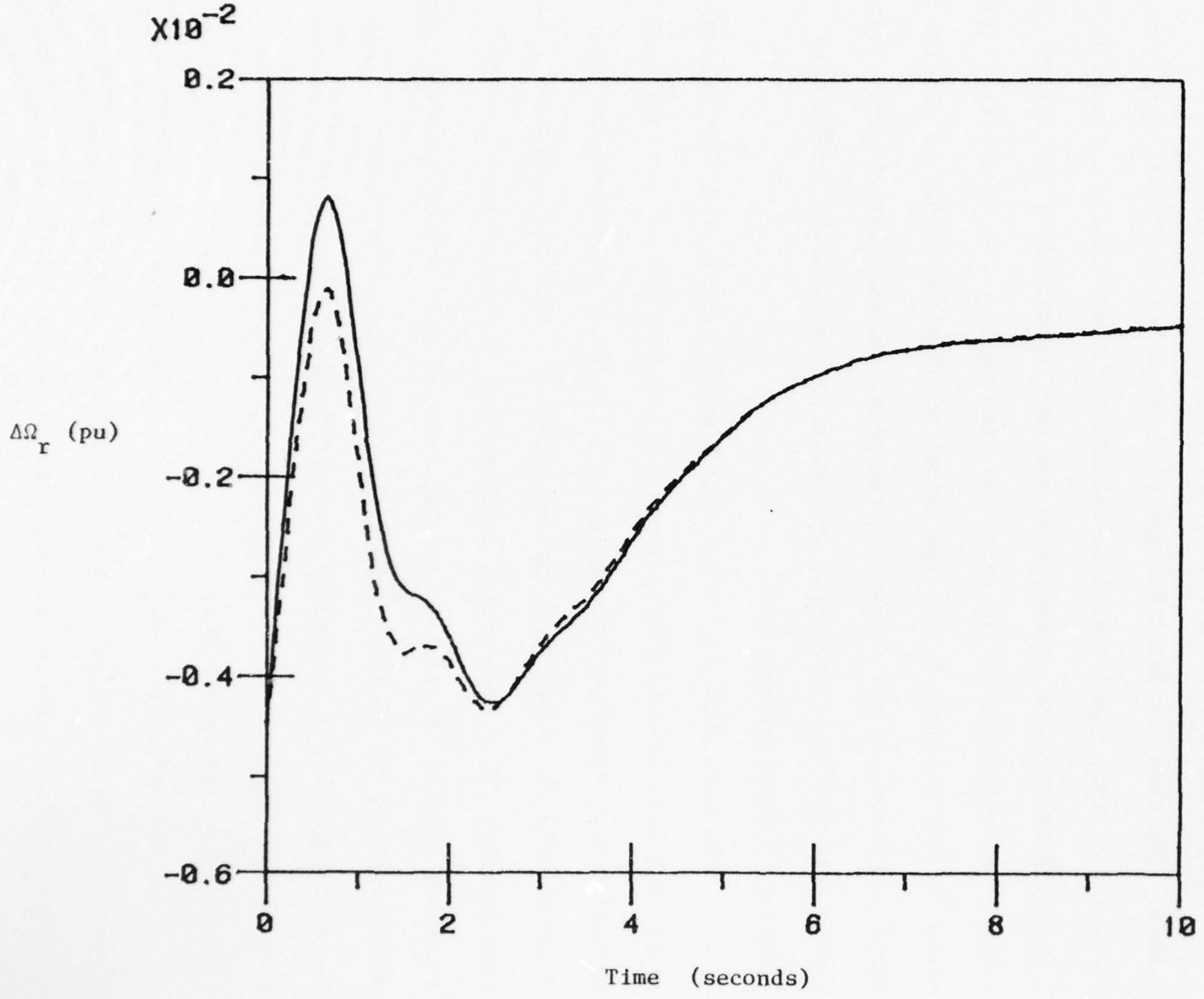


Fig. 4.3. Plots of exact and approximate $\Delta\Omega_r$.

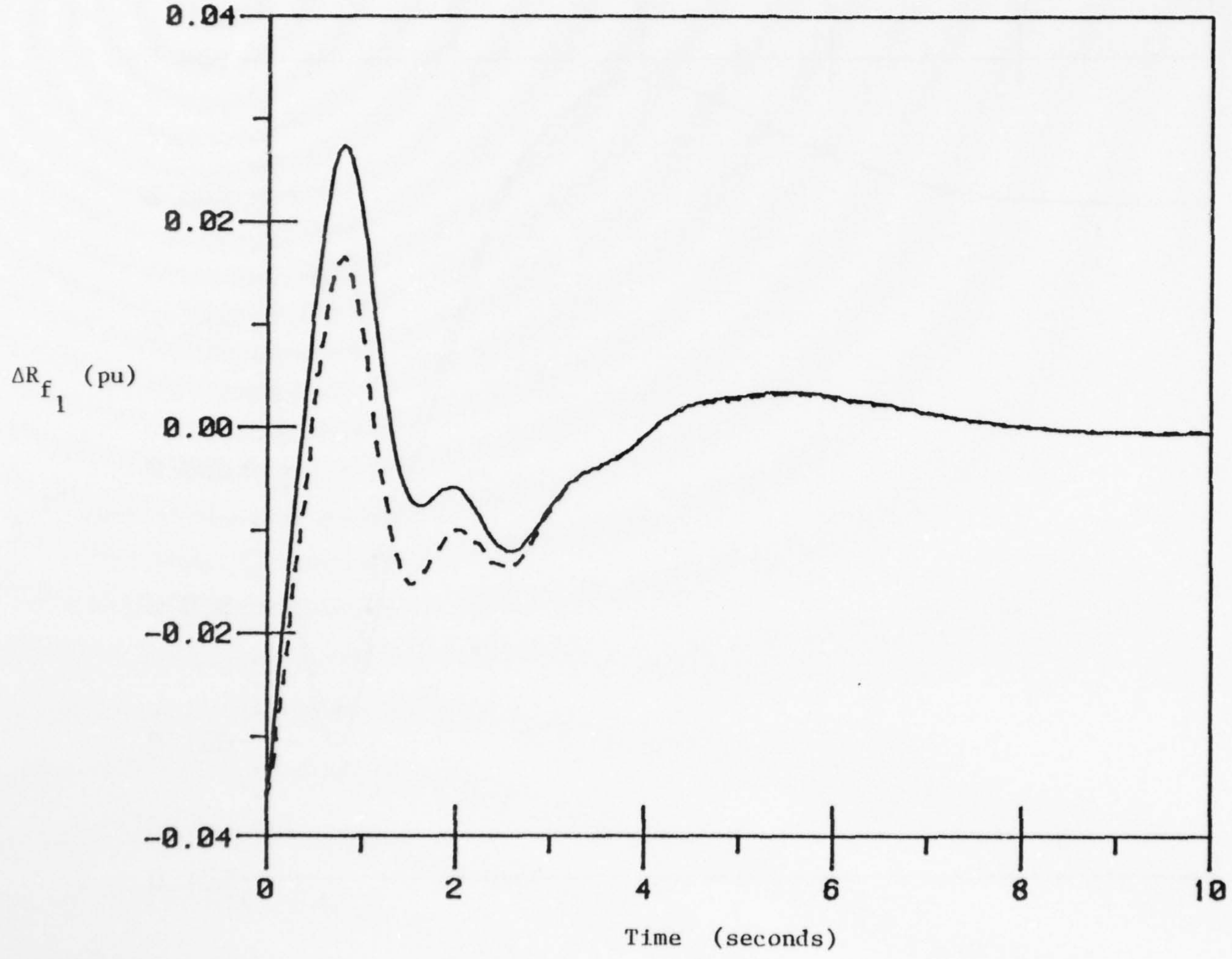


Fig. 4.4. Plots of exact and approximate ΔR

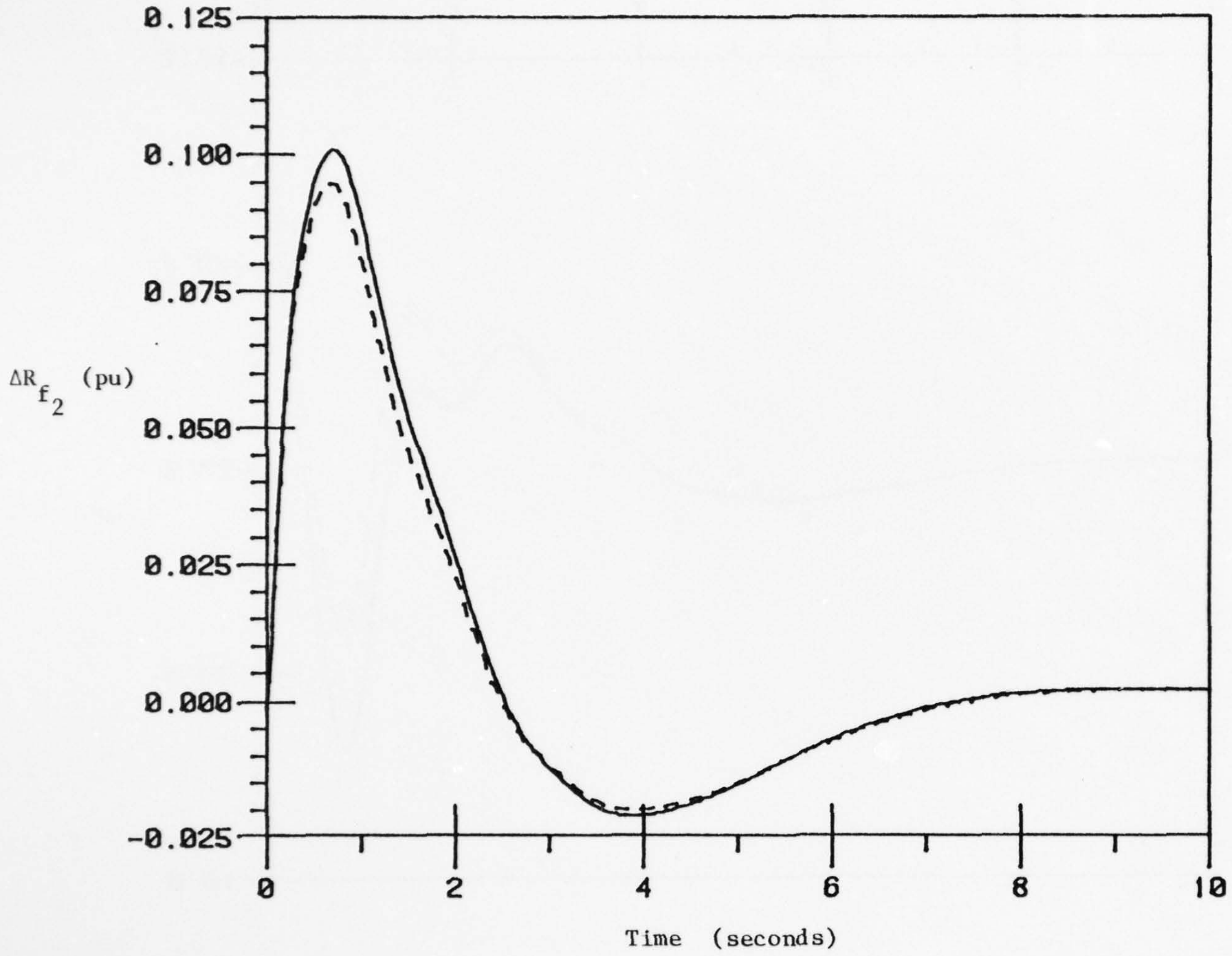


Fig. 4.5. Plots of exact and approximate ΔR_{f_2}.

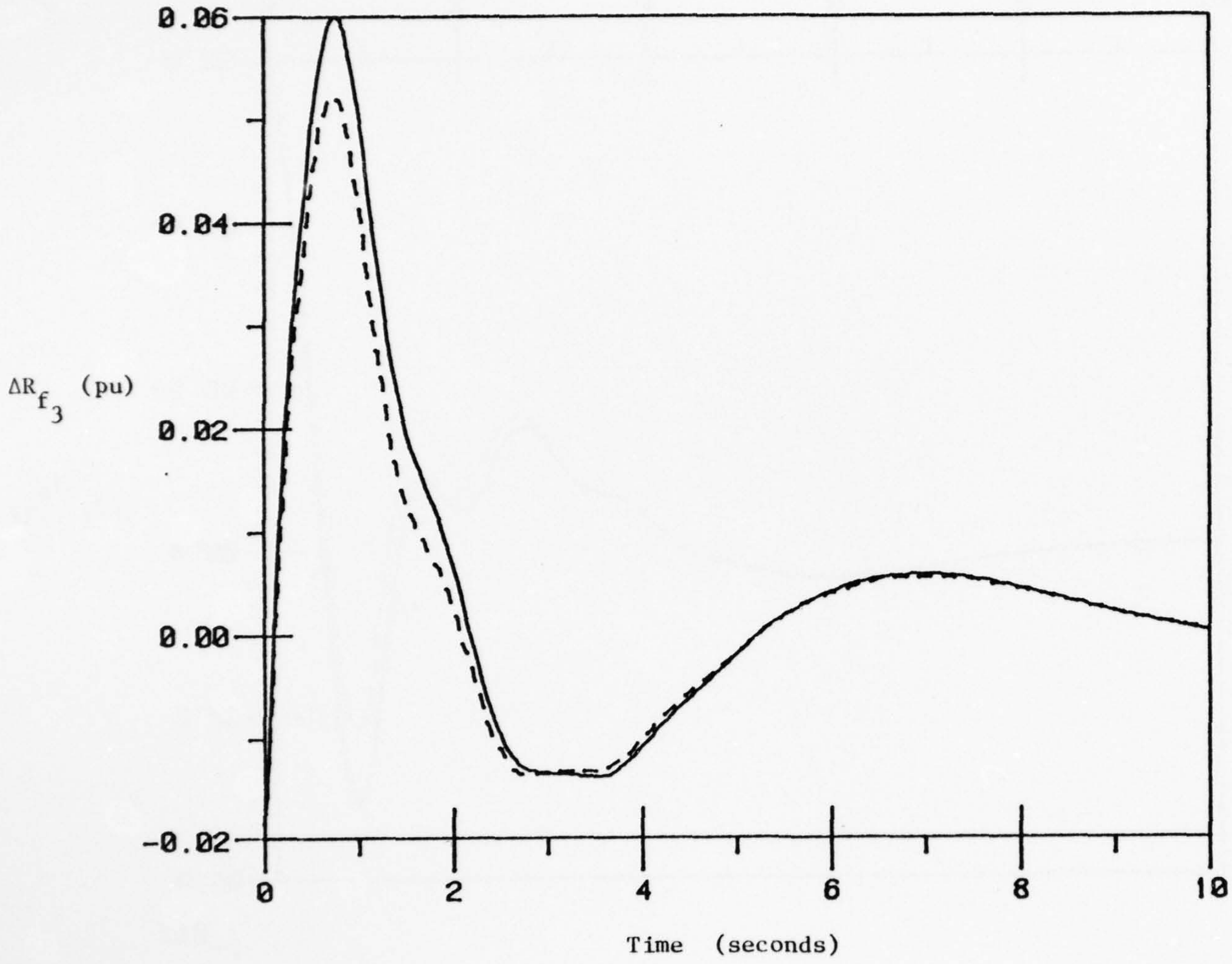


Fig. 4.6. Plots of exact and approximate ΔR_{f_3}.

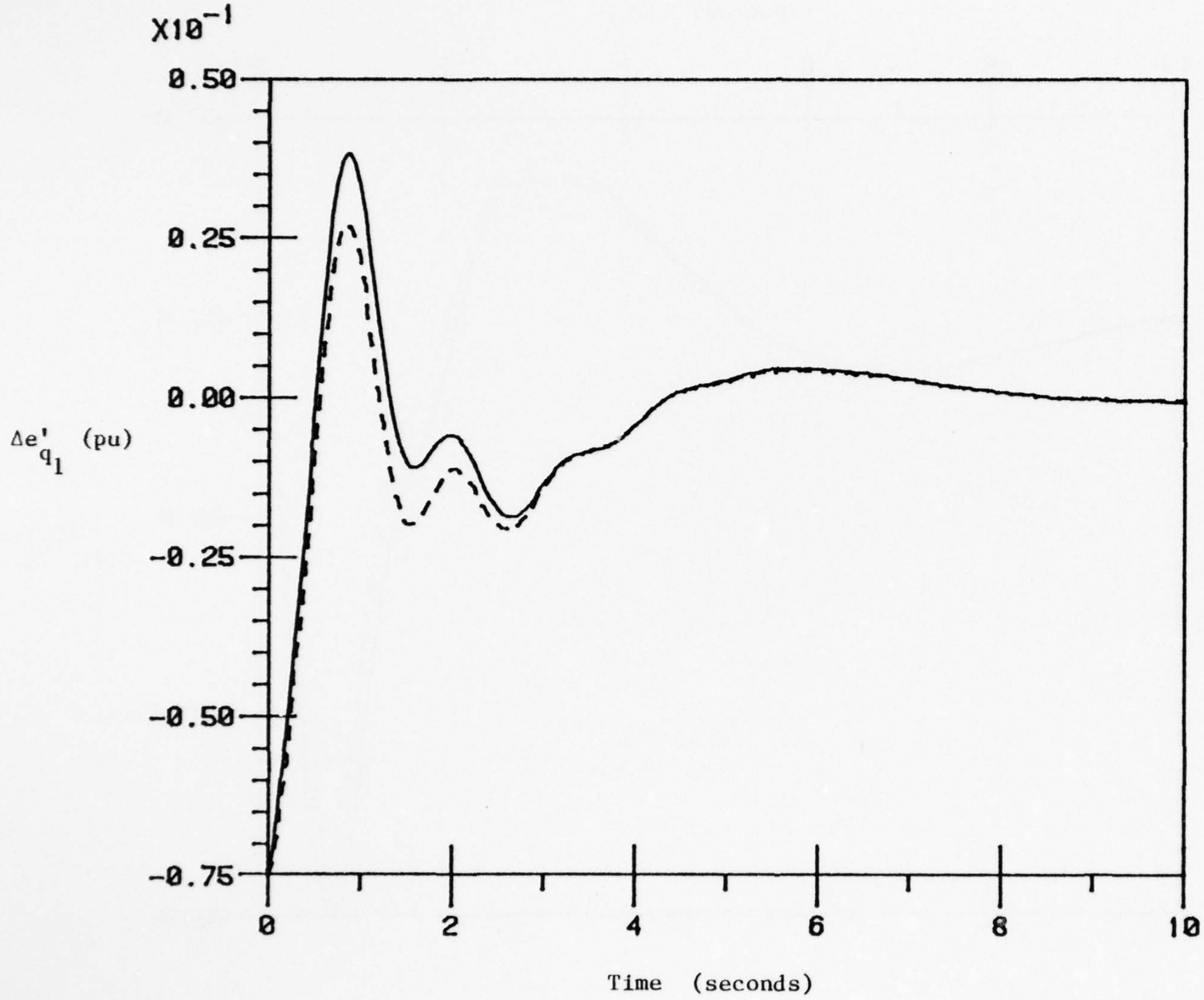


Fig. 4.7. Plots of exact and approximate $\Delta e'_{q_1}$.

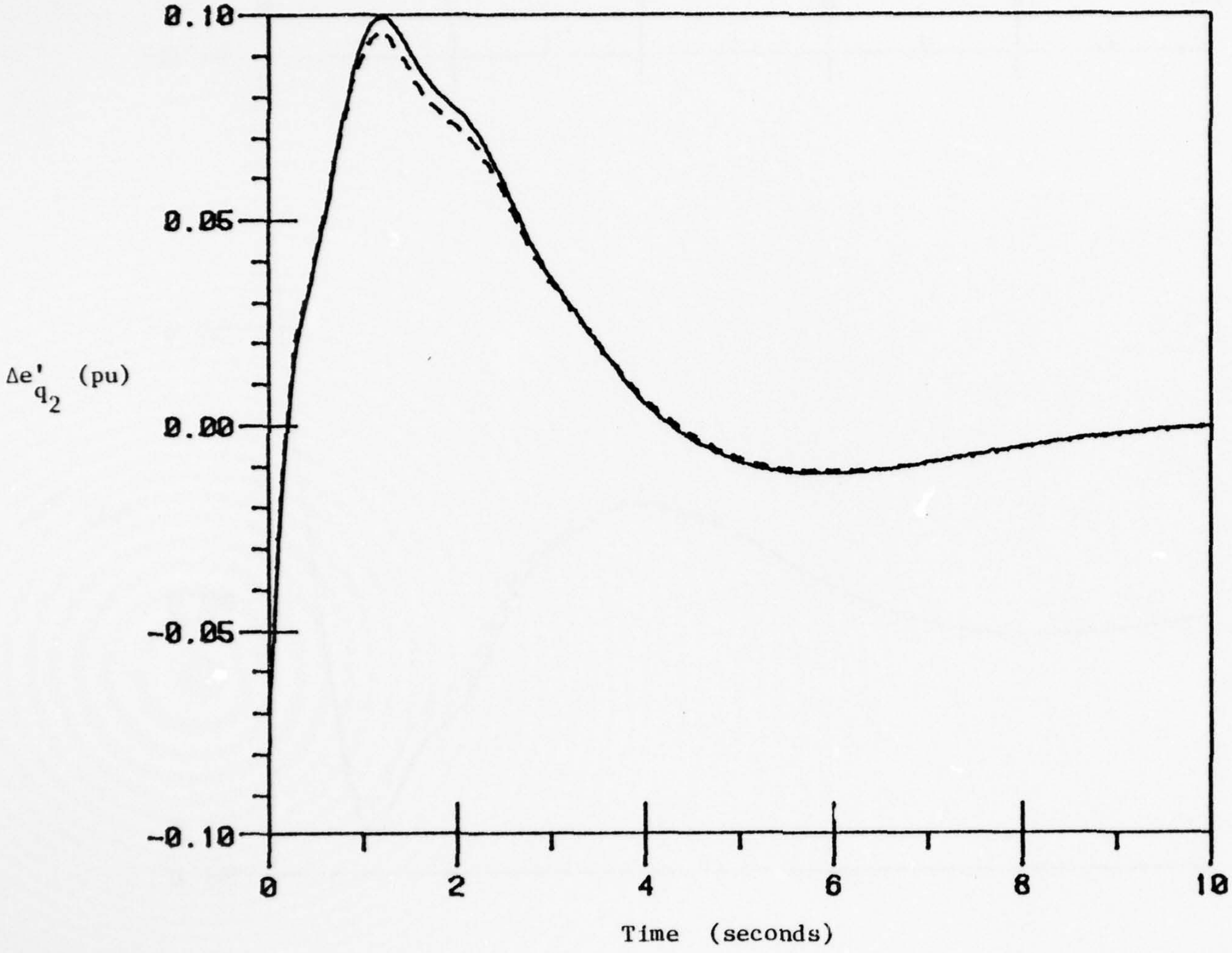


Fig. 4.8. Plots of exact and approximate $\Delta e'_{q_2}$.

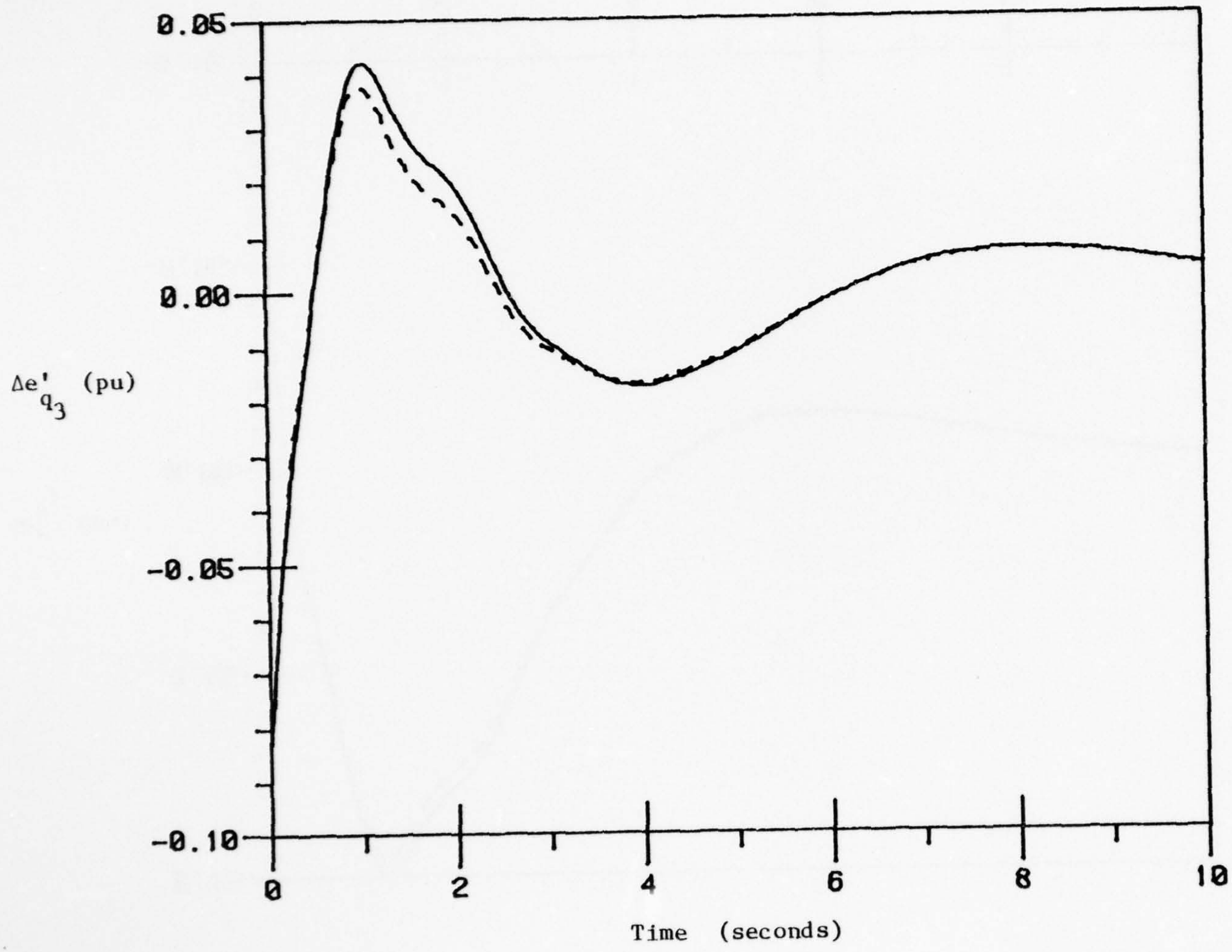


Fig. 4.9. Plots of exact and approximate $\Delta e'_{q_3}$.

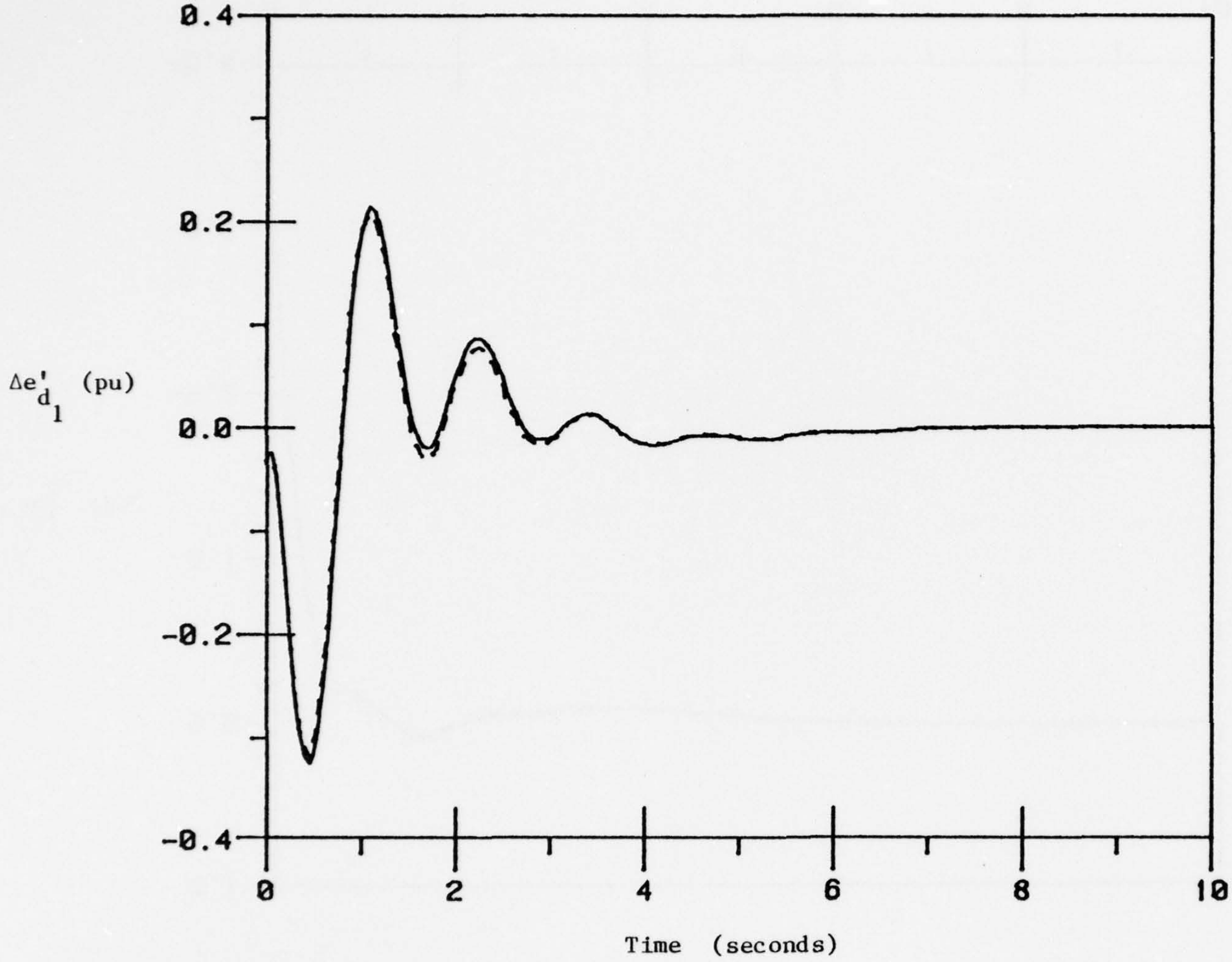


Fig. 4.10. Plots of exact and approximate $\Delta e'_{d_1}$.

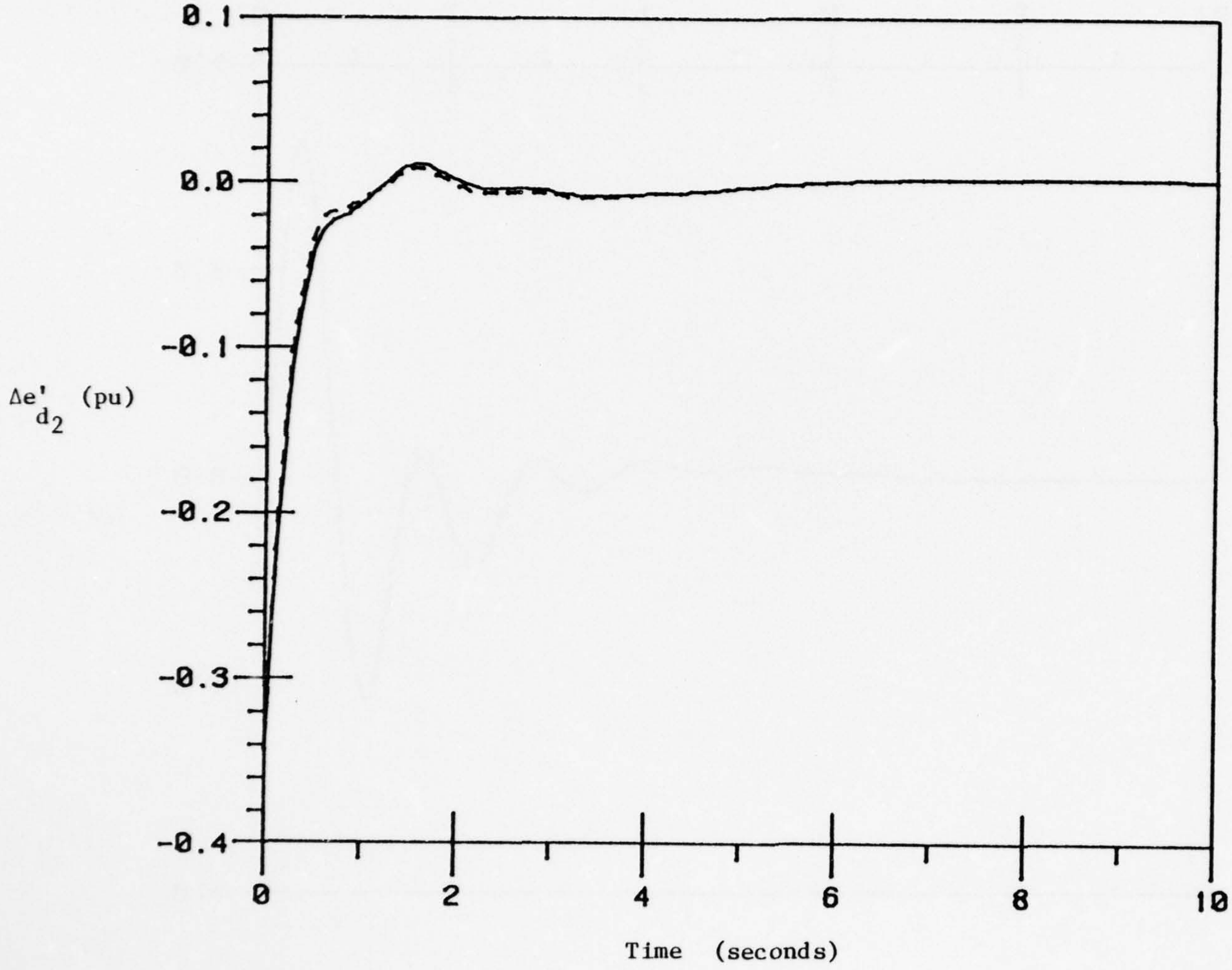


Fig. 4.11. Plots of exact and approximate $\Delta e'_{d_2}$.

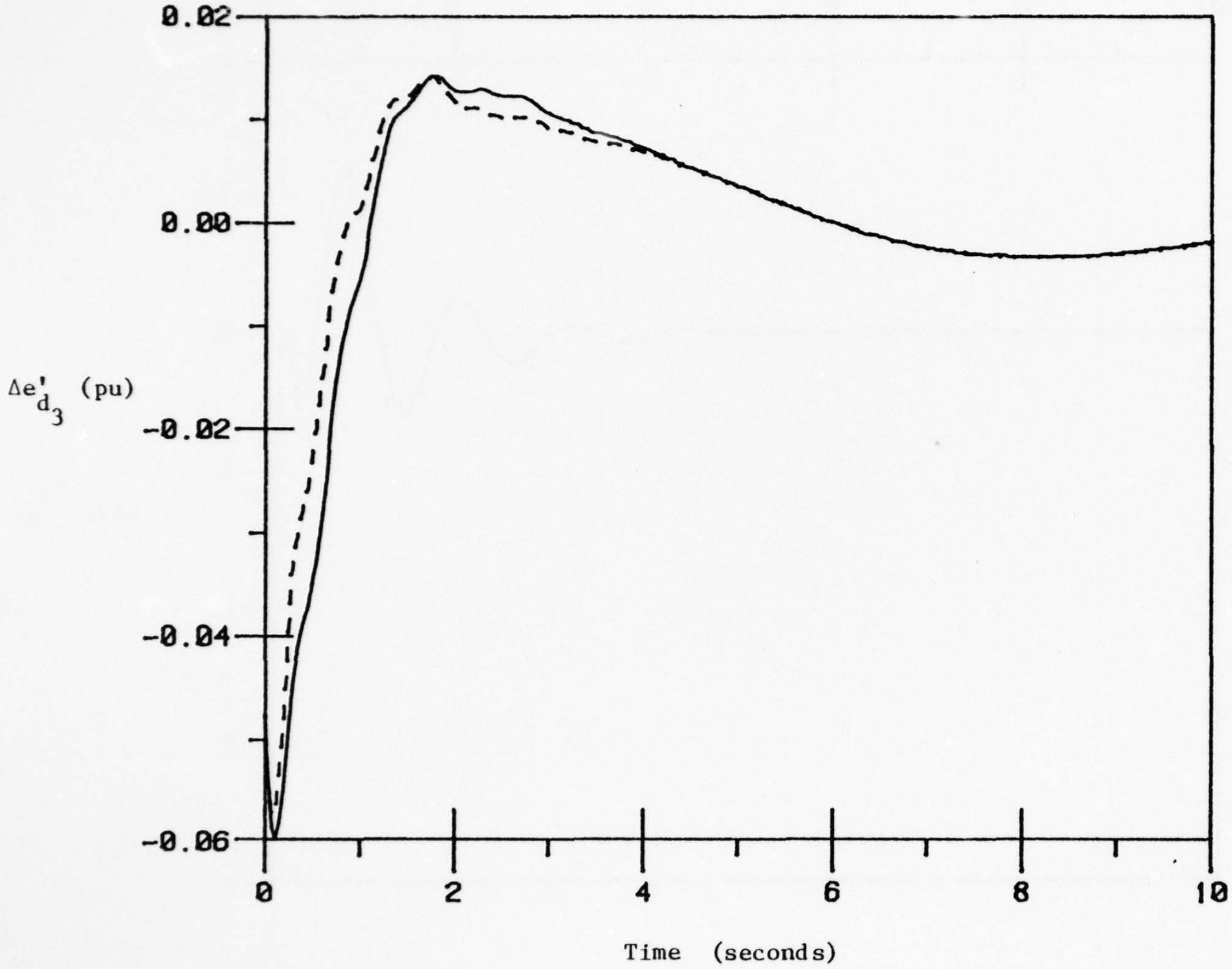


Fig. 4.12. Plots of exact and approximate $\Delta e'_{d_3}$.

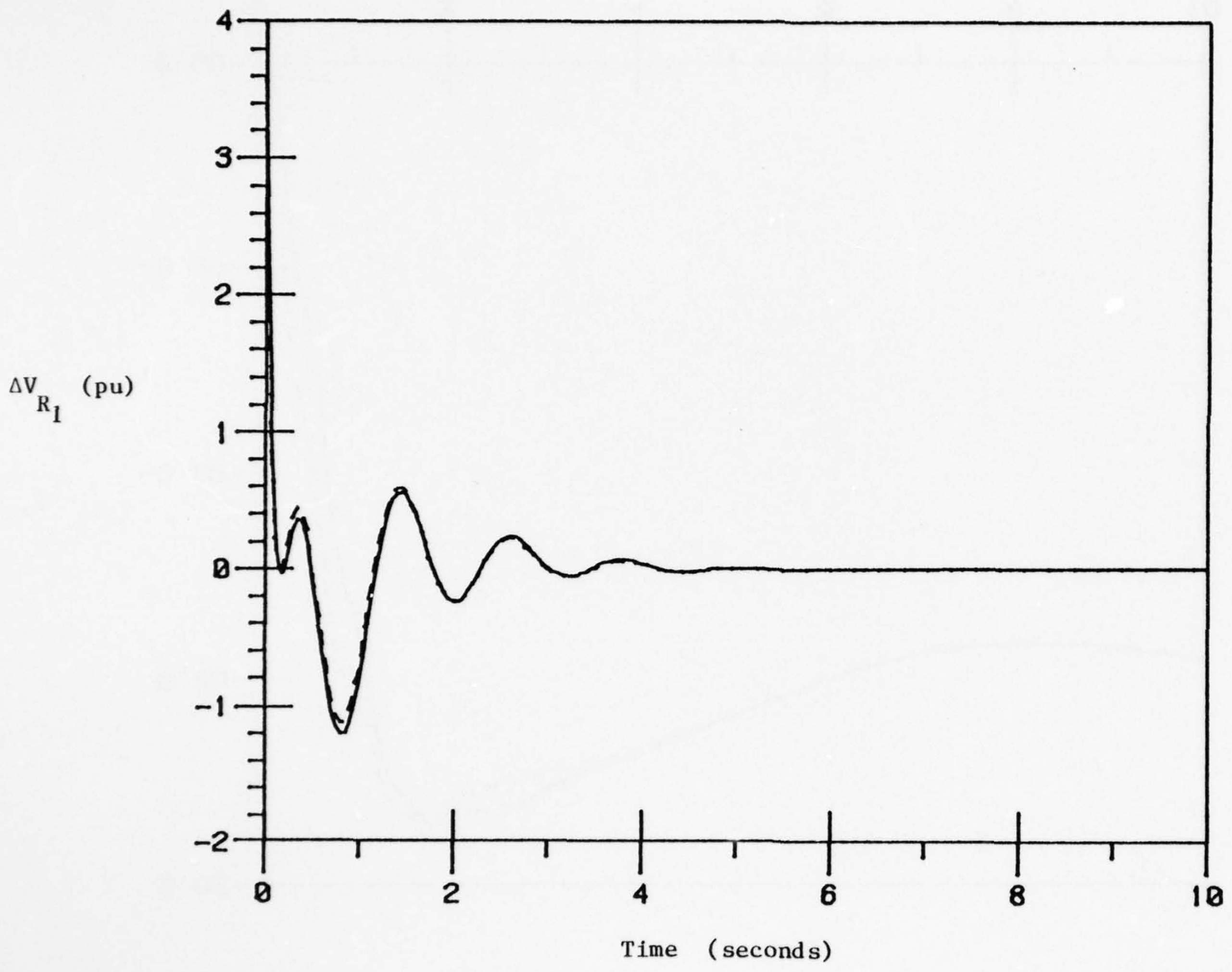


Fig. 4.13. Plots of exact and approximate ΔV_{R_1}.

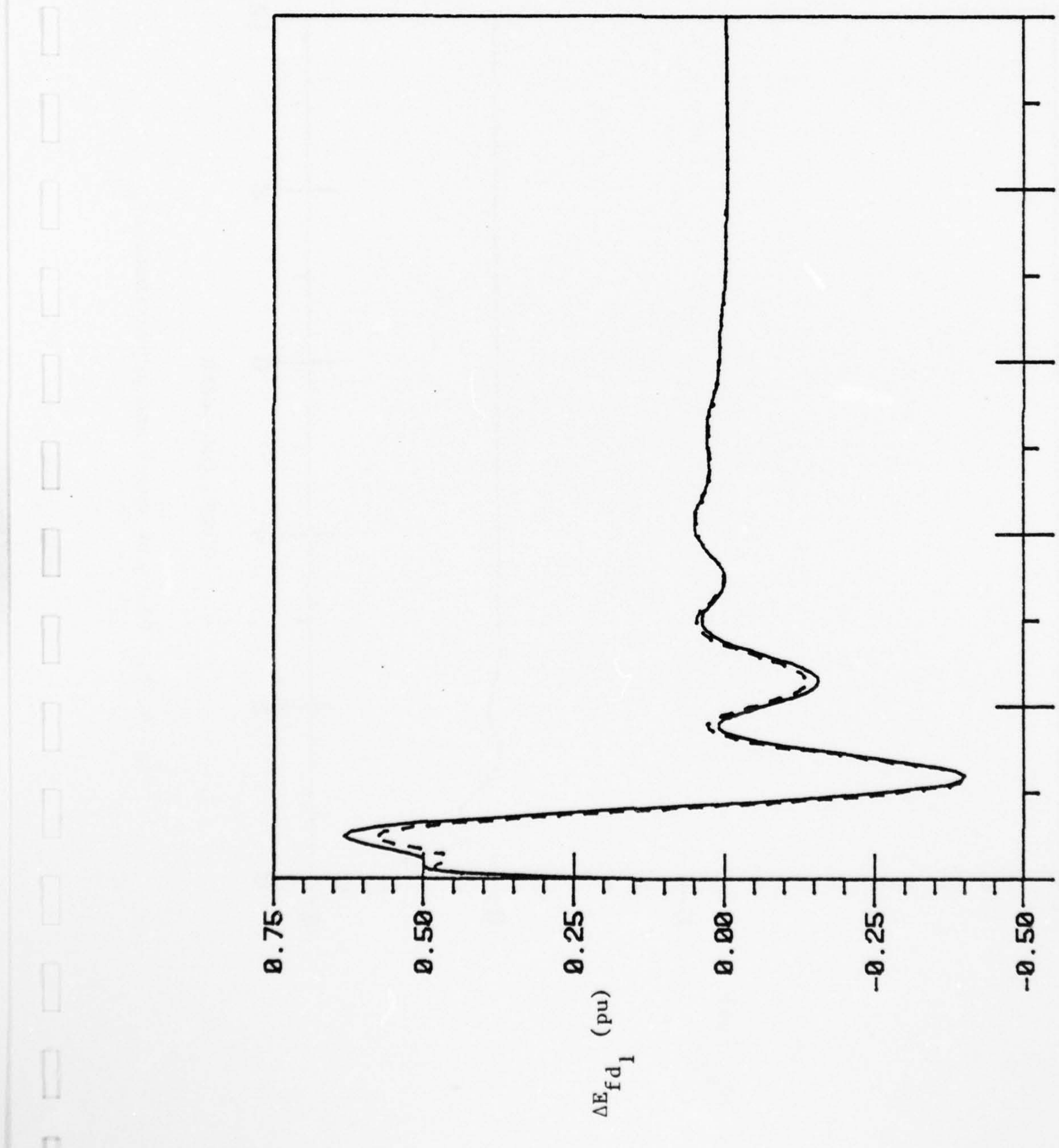


Fig. 4.14. Plots of exact and approximate ΔE_{fd_1}.

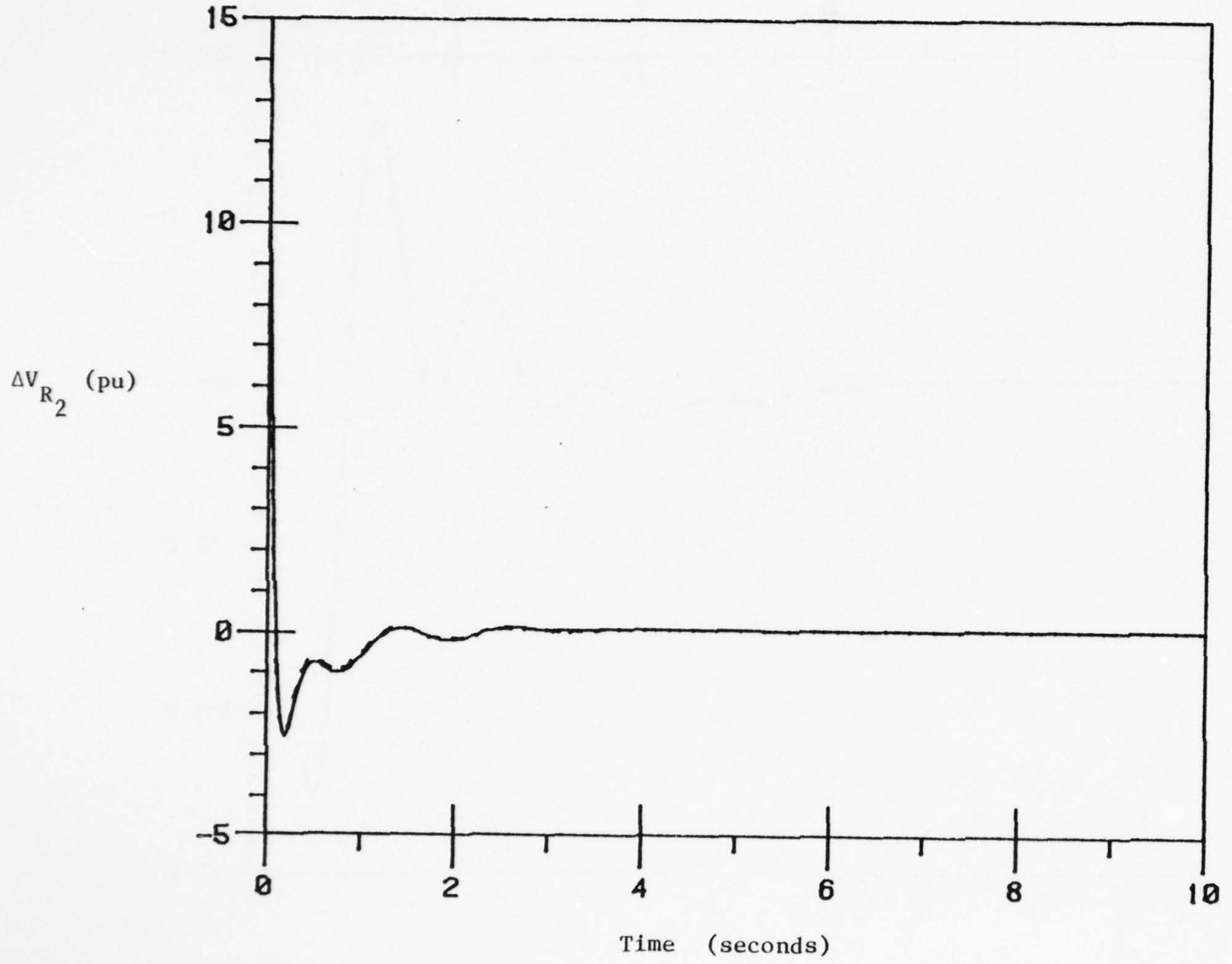


Fig. 4.15. Plots of exact and approximate ΔV_{R_2}.

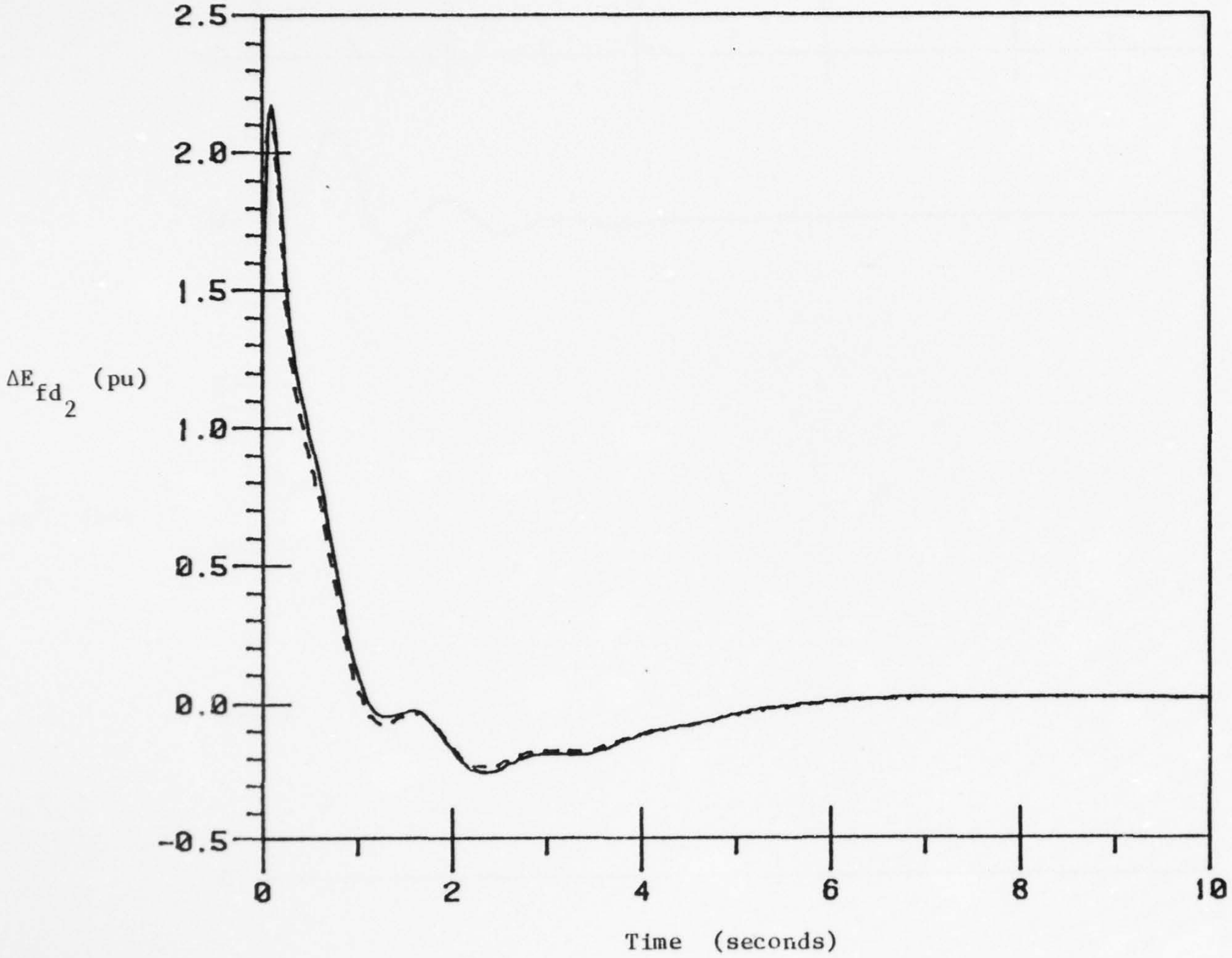


Fig. 4.16. Plots of exact and approximate ΔE_{fd_2}.

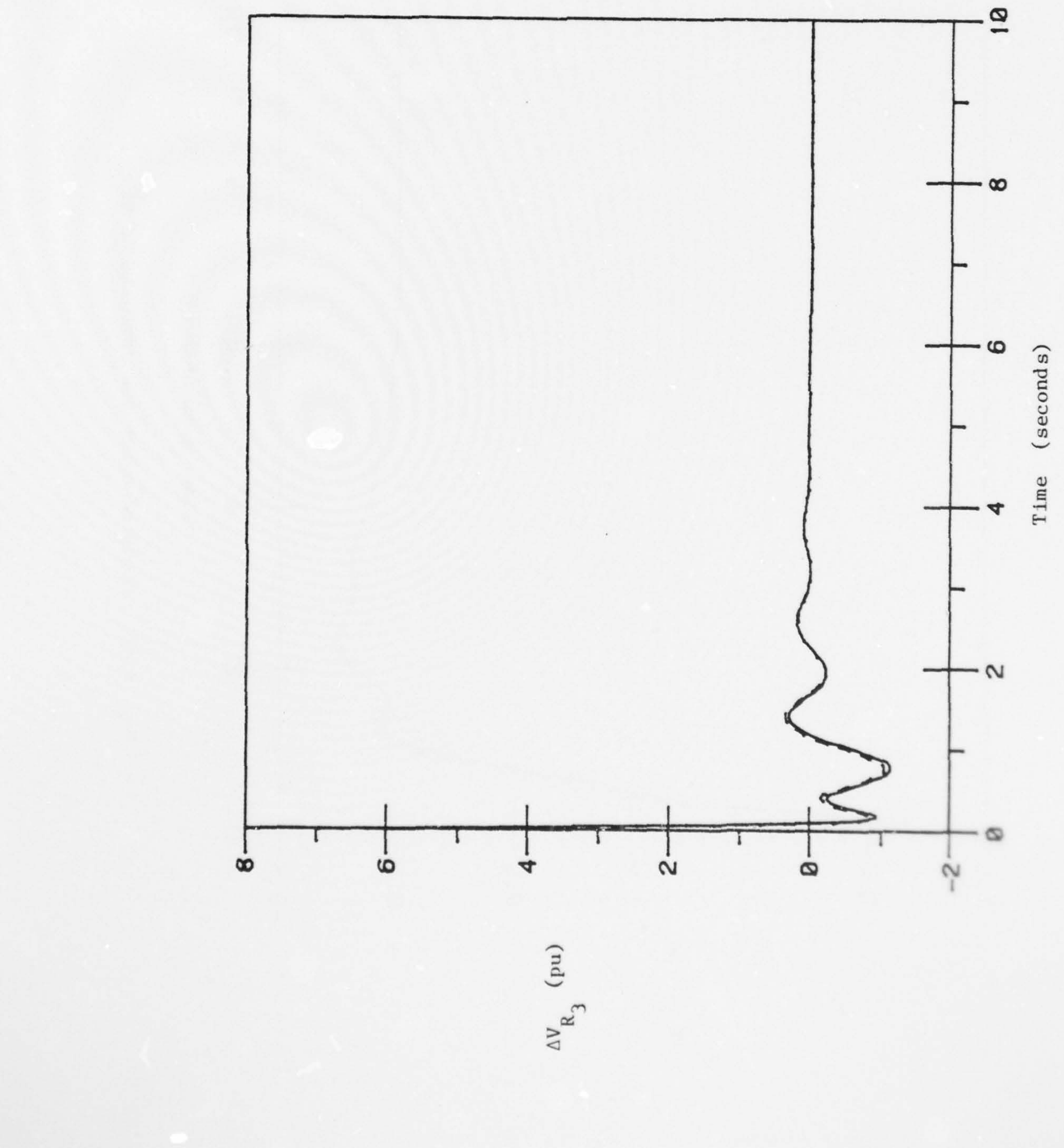
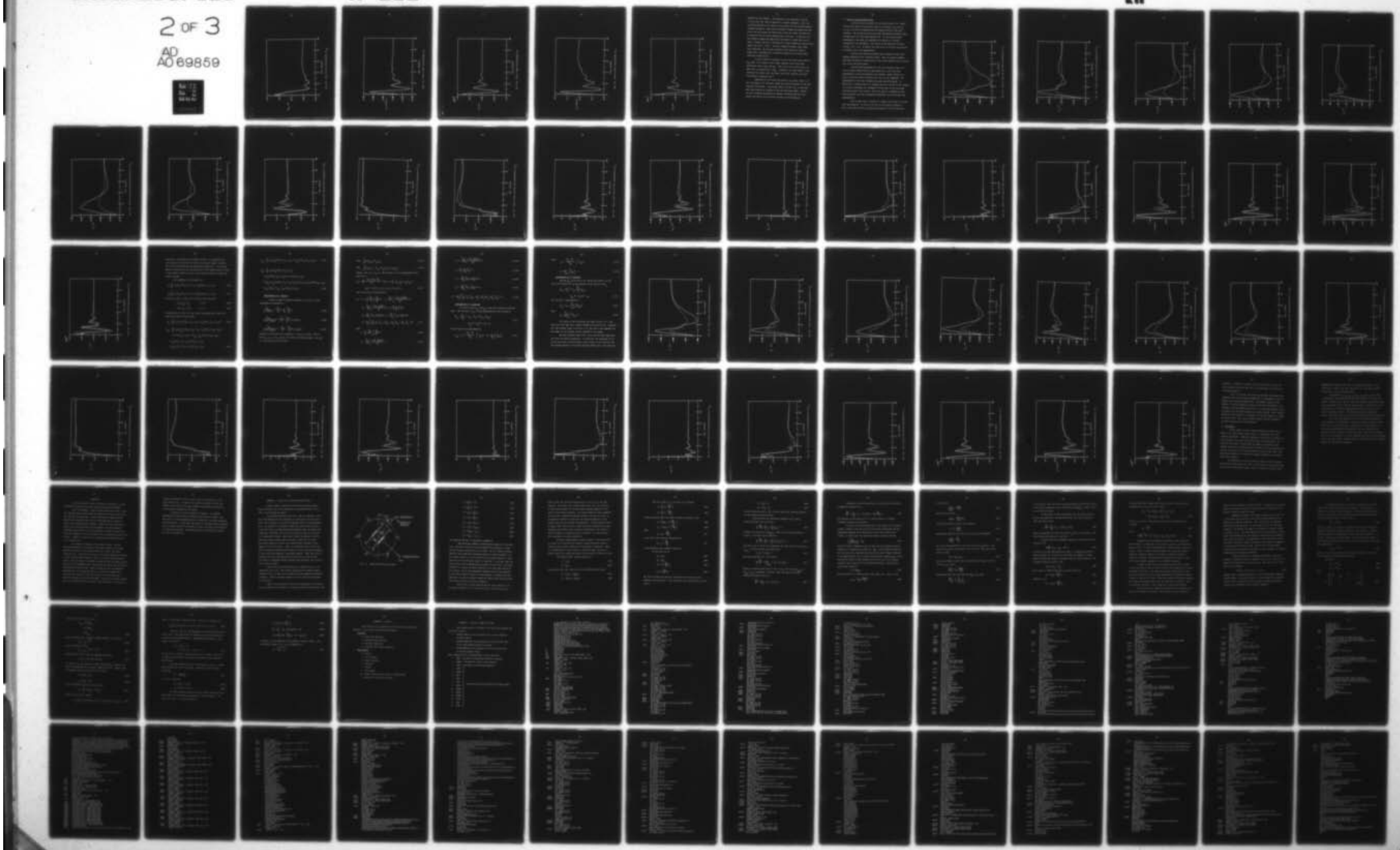
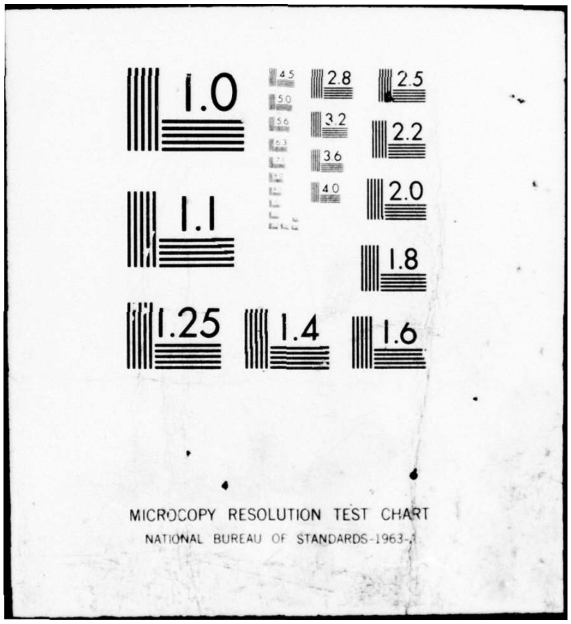


Fig. 4.17. Plots of exact and approximate ΔV_{R_3}.

AD-A069 859 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/3
A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS.(U)
AUG 78 J J ALLEMONG DAAB07-72-C-259
UNCLASSIFIED R-818 NL

2 OF 3
AD A069859





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

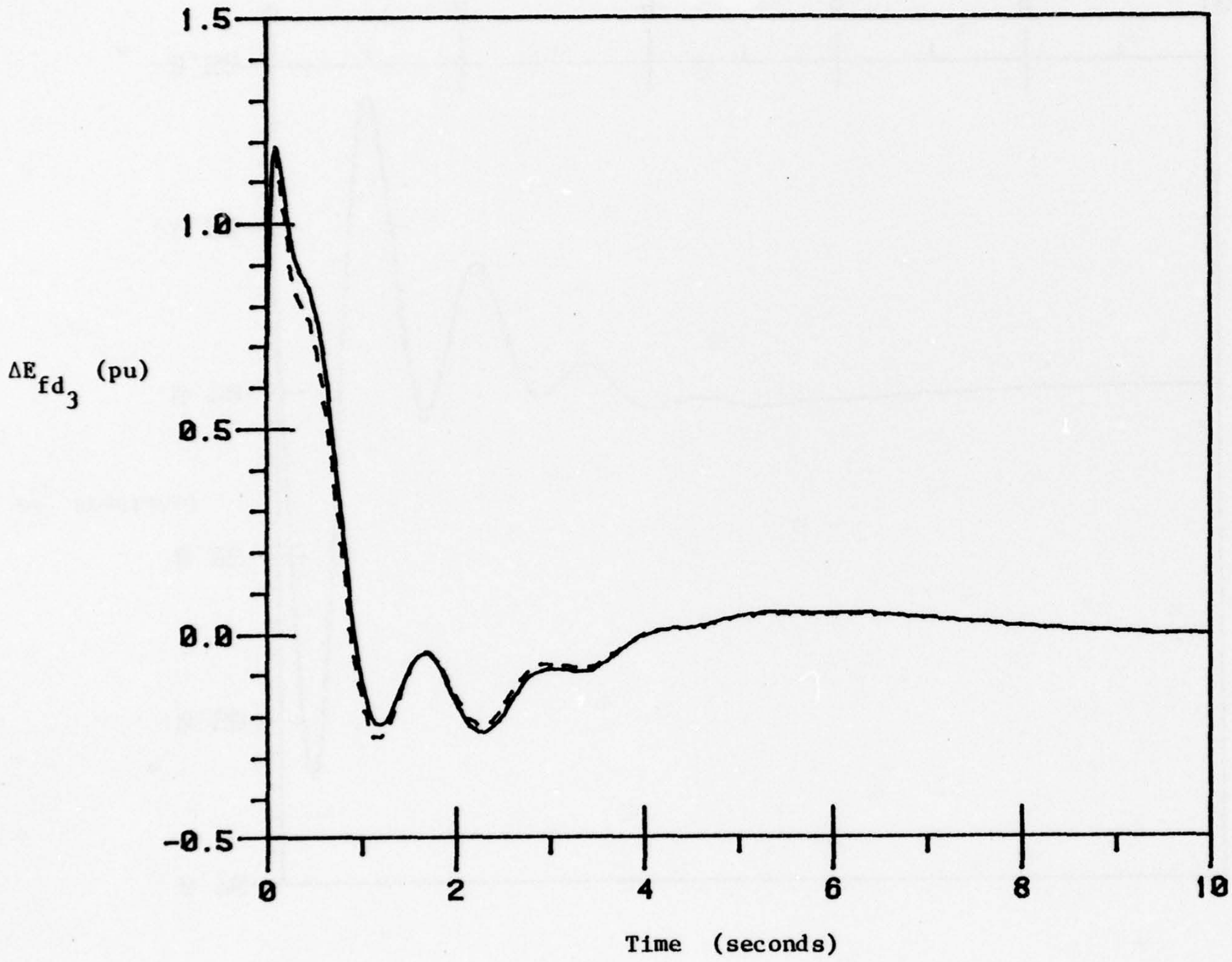


Fig. 4.18. Plots of exact and approximate ΔE_{fd_3}.

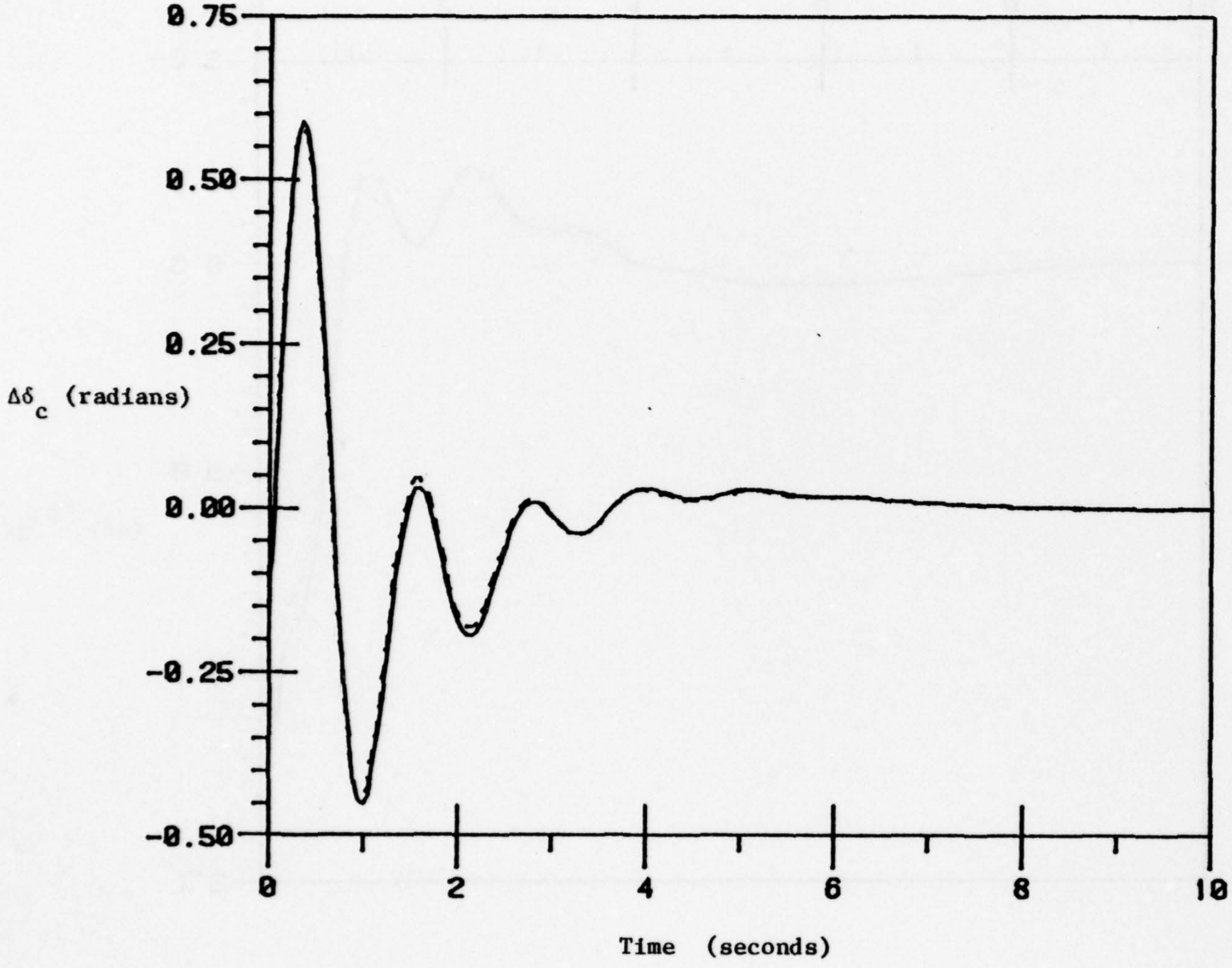


Fig. 4.19. Plots of exact and approximate $\Delta\delta_c$.

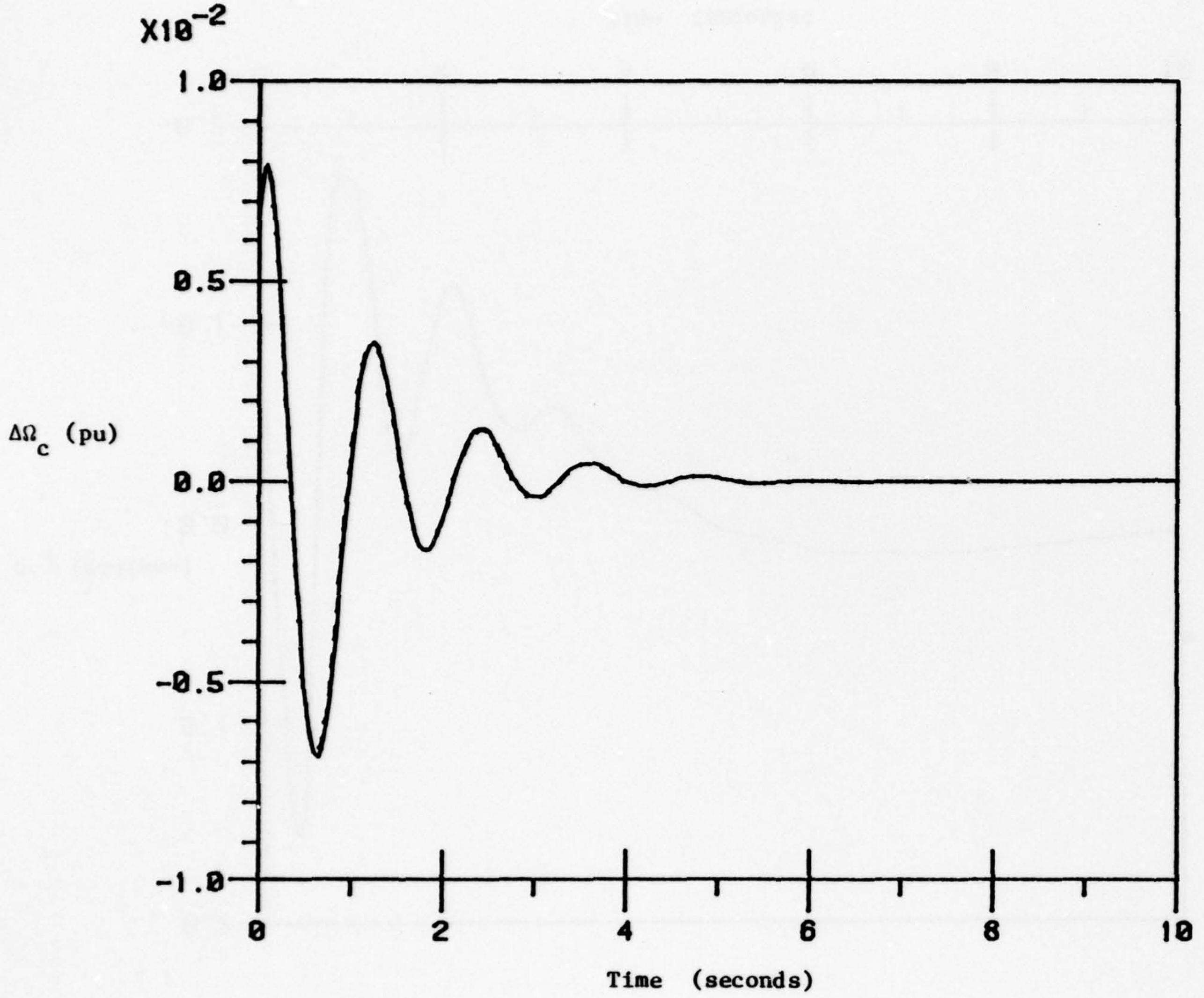


Fig. 4.20. Plots of exact and approximate $\Delta\Omega_c$.

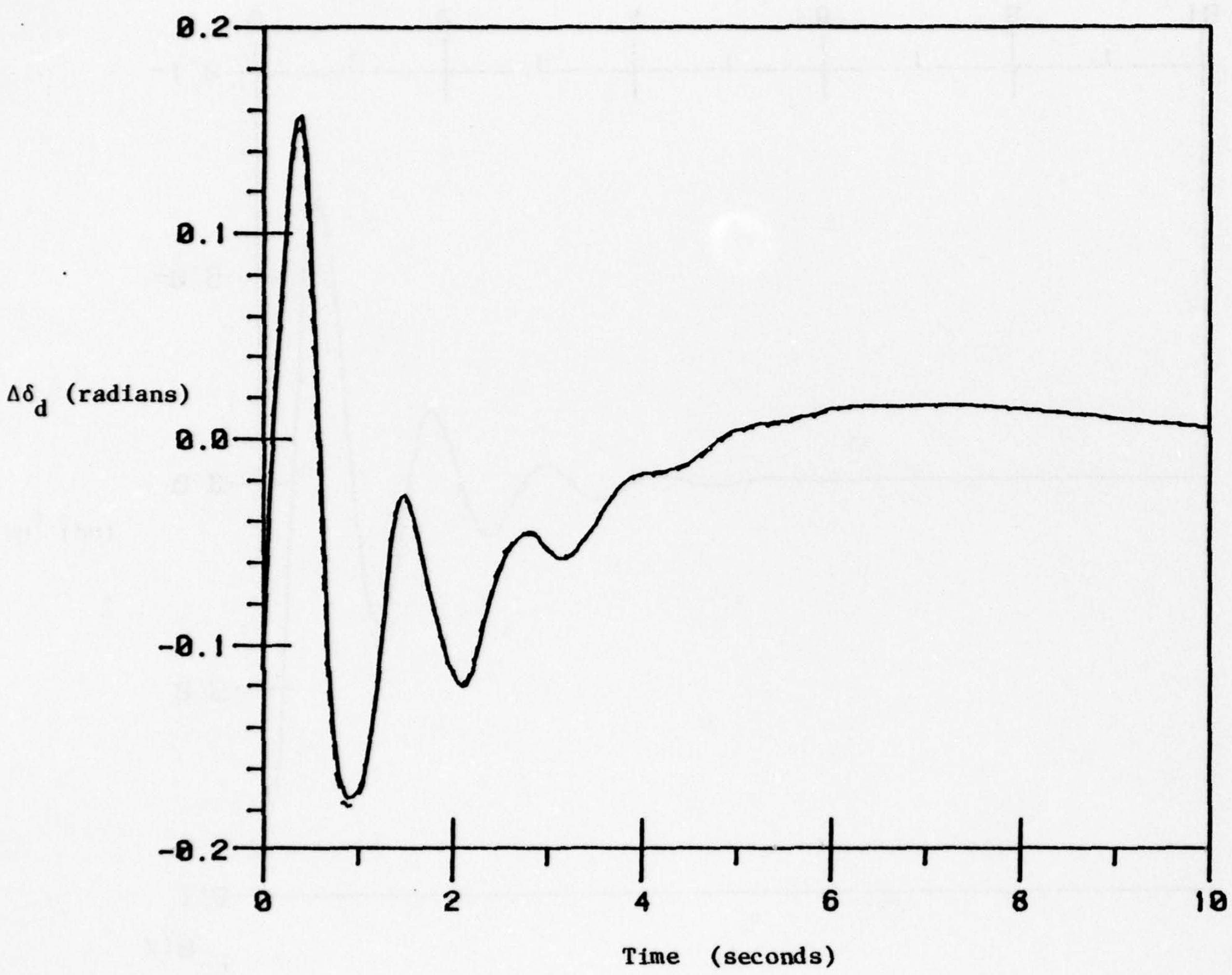


Fig. 4.21. Plots of exact and approximate $\Delta\delta_d$.

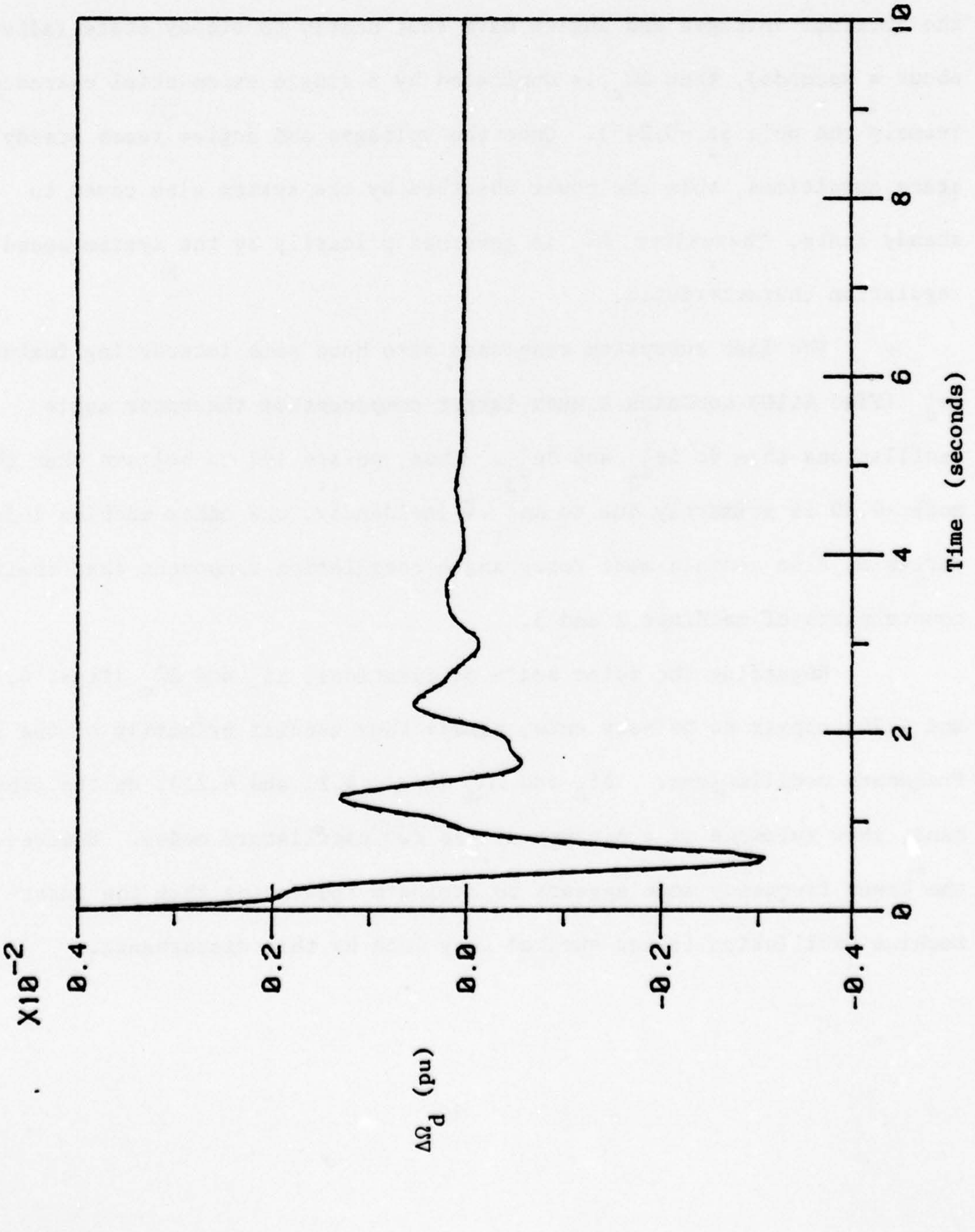


Fig. 4.22. Plots of exact and approximate $\Delta\Omega_d$.

dominated by fast dynamics. This behavior can be explained as follows. In this study, the loads are modelled by constant impedances. Thus, the real power absorbed by the system is very sensitive to the internal machine voltages and angles. Since both the internal voltages and angles have fast parts, the total system load should have a fast part; hence, we should not be surprised that the system frequency has a fast part. In addition, once the internal voltages and angles have come nearly to steady state (after about 4 seconds), then $\Delta\Omega_r$ is dominated by a single exponential characteristic (namely the pole at -0.245). Once the voltages and angles reach steady state conditions, then the power absorbed by the system also comes to steady state. Thereafter, $\Delta\Omega_r$ is governed primarily by the system speed regulation characteristic.

The fast subsystem responses also have some interesting features. $\Delta e'_{d_1}$ (Fig. 4.10) contains a much larger component of the rotor angle oscillations than do $\Delta e'_{d_2}$ and $\Delta e'_{d_3}$. Thus, we are led to believe that the mode -6.90 is primarily due to $\Delta e'_{d_1}$. Incidently, the other machine 1 fast variables also contain more rotor angle oscillation component than their counterparts of machines 2 and 3.

Regarding the rotor angle oscillations, $\Delta\delta_c$ and $\Delta\Omega_c$ (Figs. 4.19 and 4.20) appear to be very pure, namely they consist primarily of the lower frequency oscillations. $\Delta\delta_d$ and $\Delta\Omega_d$ (Figs. 4.21 and 4.22), on the other hand, show evidence of a mixture of the two oscillatory modes. However, the lower frequency mode appears to dominate indicating that the inter-machine oscillation is not excited very much by this disturbance.

4.5 Study of the Non-Linear System

In this section we consider the non-linear system (4.5). Based on the linear study of the previous section, we choose Ω_r, R_{f_i}, and e'_{q_i} $(i = 1,2,3)$ as the slow variables and the remaining states as the fast variables. The calculation of the zero-order approximation proceeds along the same lines as in the single machine case. To test the zero-order approximation, the system was simulated for a period of 10 seconds following the test disturbance. The results of the simulation are shown in Figs. 4.23 - 4.42. As before, the solid curve is the exact solution and the dashed curve is the approximation.

We notice the similarity between these responses and the corresponding responses of the linearized system. Thus, the general comments made about the physical interpretation of the linear responses can be carried over to the non-linear system.

The zero-order approximation of the slow variables (Figs. 4.23 - 4.29) evidently suffers from the same difficulties as the zero-order approximation of the single machine slow variables, namely, absence of a shift of the slow initial conditions and a lack of fast components. The approximations of the fast variables also have some shortcomings. In particular, in several cases it is apparent that the slow part of the variable is in error; furthermore the "frequency" of the slower of two rotor angle oscillation modes is not correct. Since this mode is a component of many other responses, this error propagates through most of the zero-order fast approximations.

Thus we again find it necessary to compute corrections to the zero-order approximation. To this end, we will use the method of Chapter 3. Most of the labor involved in applying this method to the slow subsystem

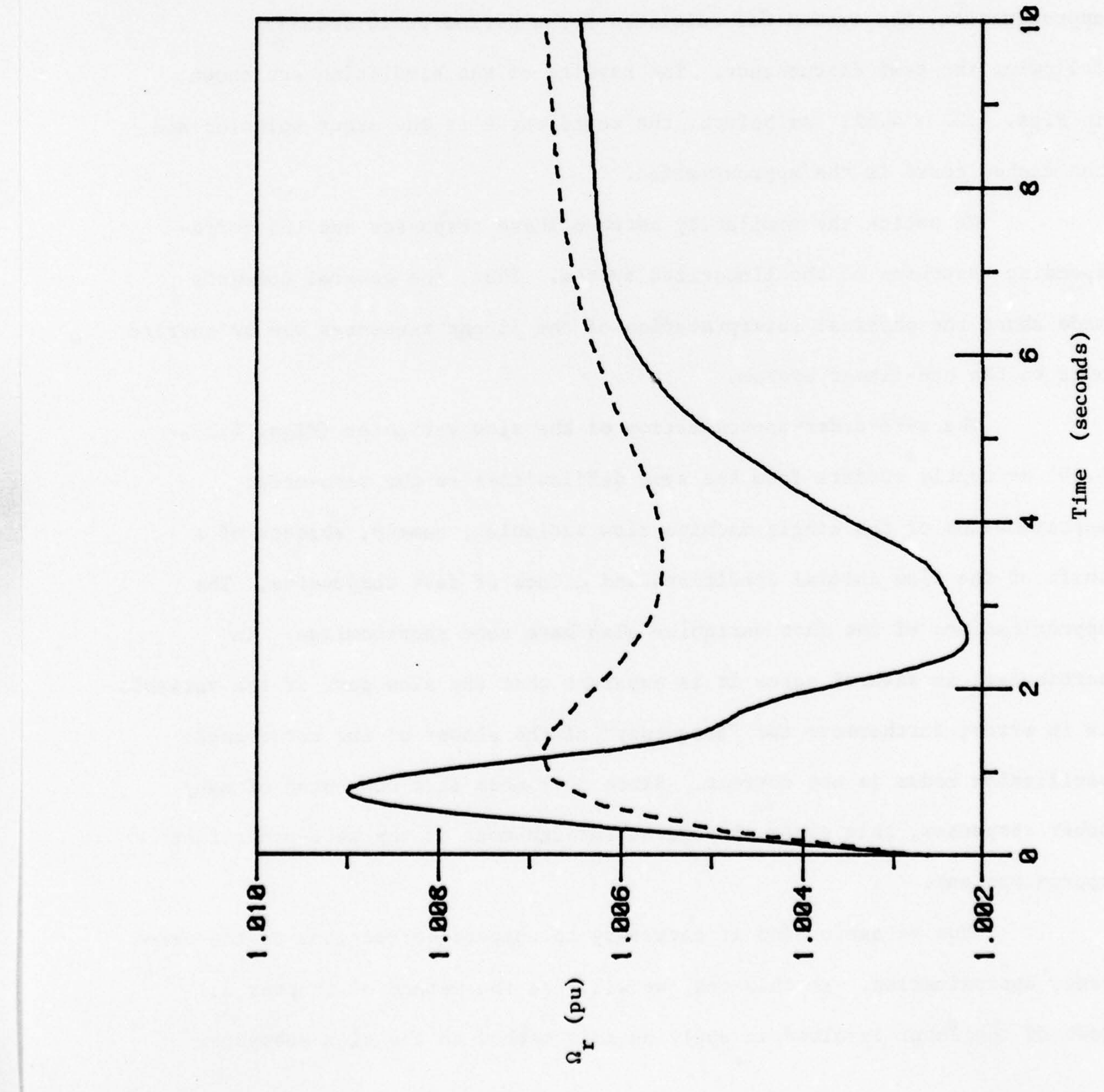


Fig. 4.23. Plots of exact and zero-order approximation of Ω_r.

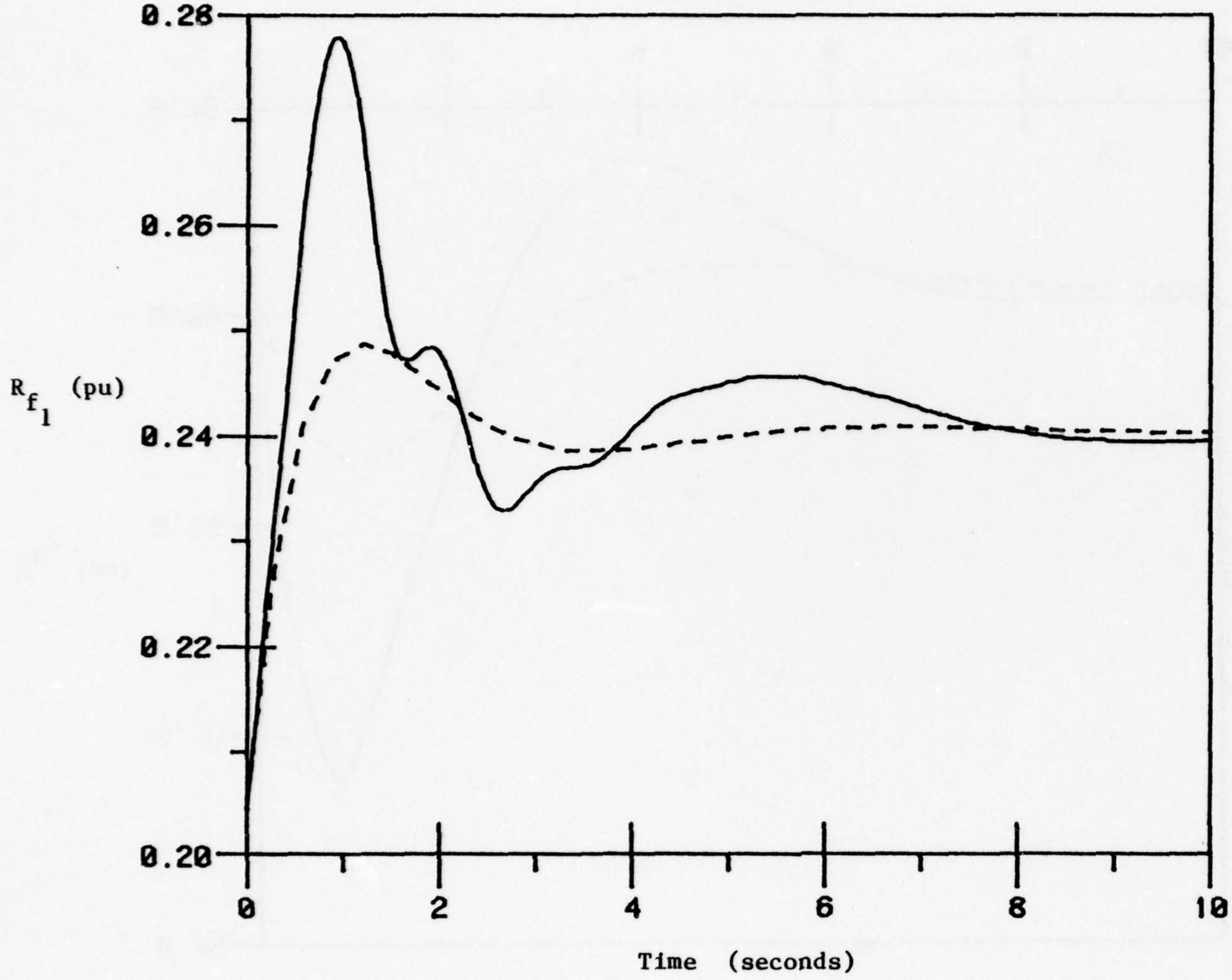


Fig. 4.24. Plots of exact and zero-order approximation of R_{f_1}.

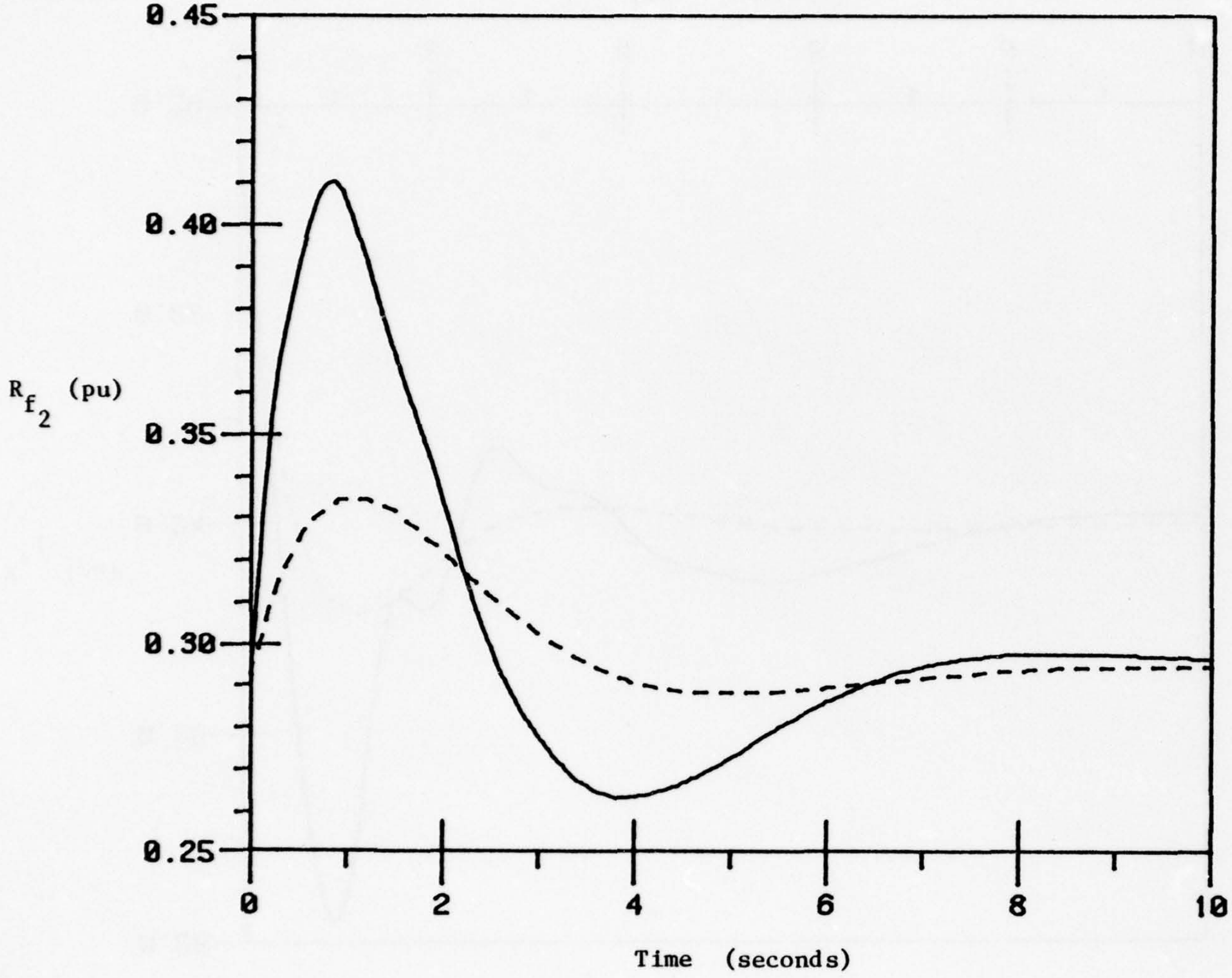


Fig. 4.25. Plots of exact and zero-order approximation of R_{f_2}.

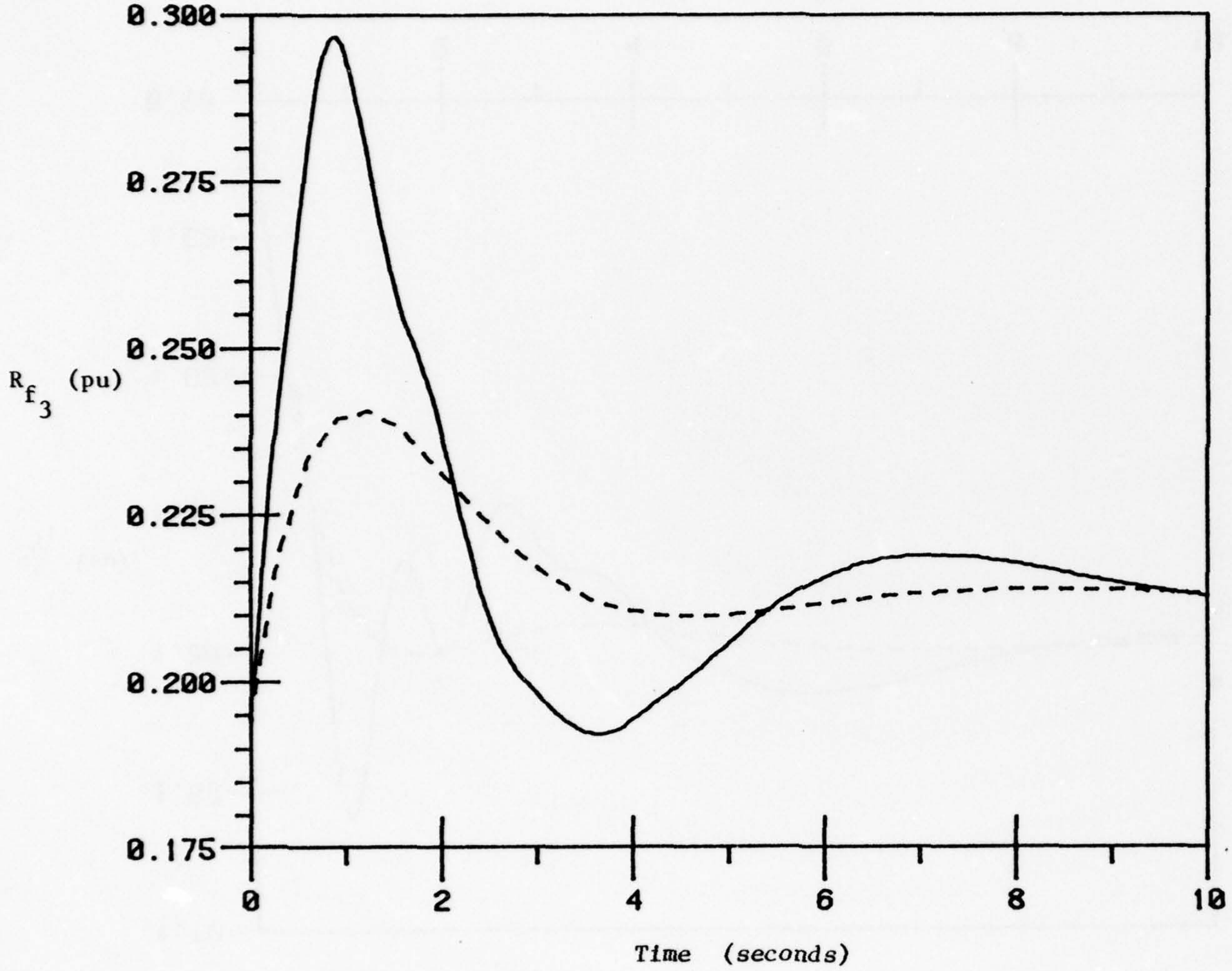


Fig. 4.26. Plcts of exact and zero-order approximation of R_{f_3}.

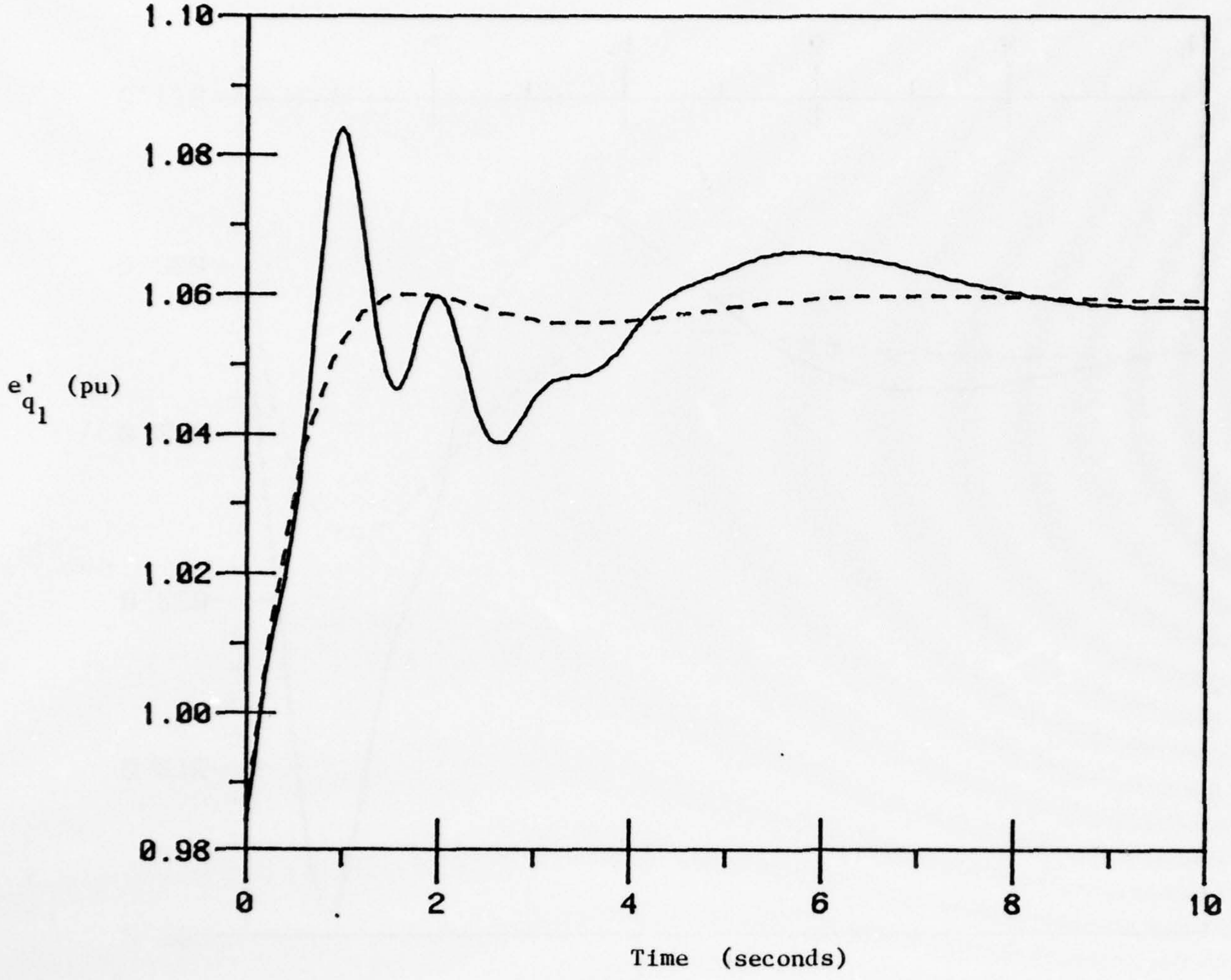


Fig. 4.27. Plots of exact and zero-order approximation of e'_{q_1}.

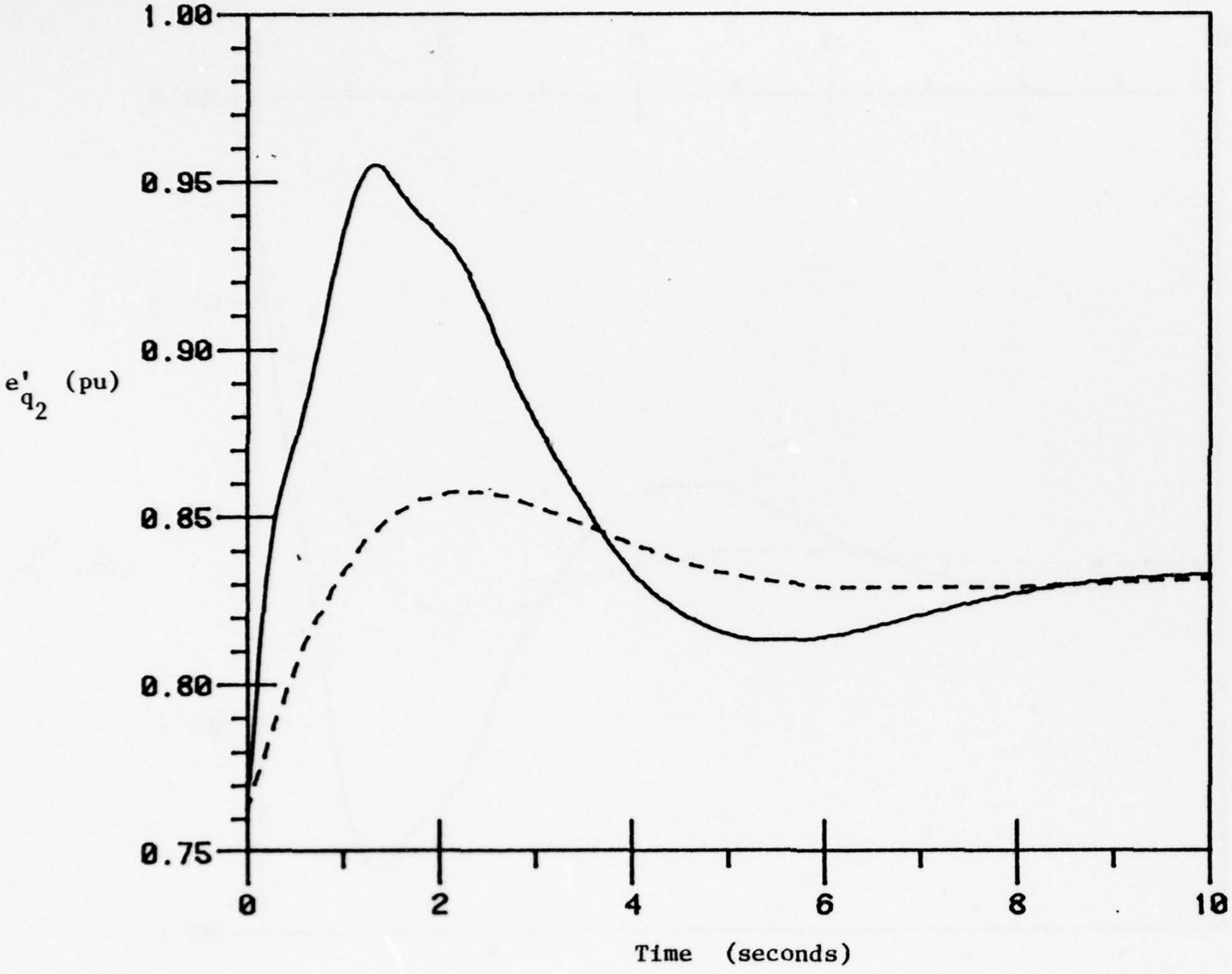


Fig. 4.28. Plots of exact and zero-order approximation of e'_{q_2}.

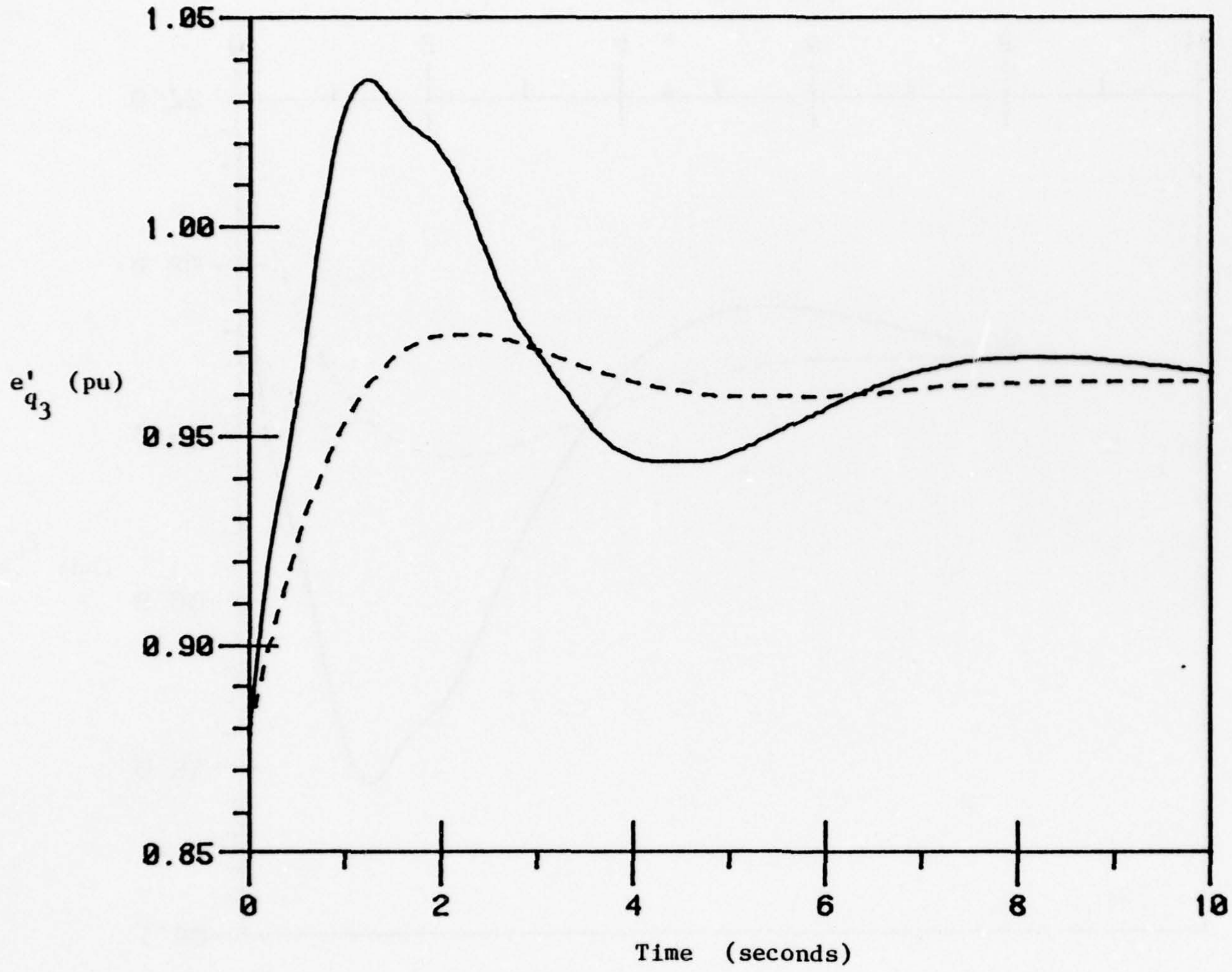


Fig. 4.29. Plots of exact and zero-order approximation of e'_{q_3}.

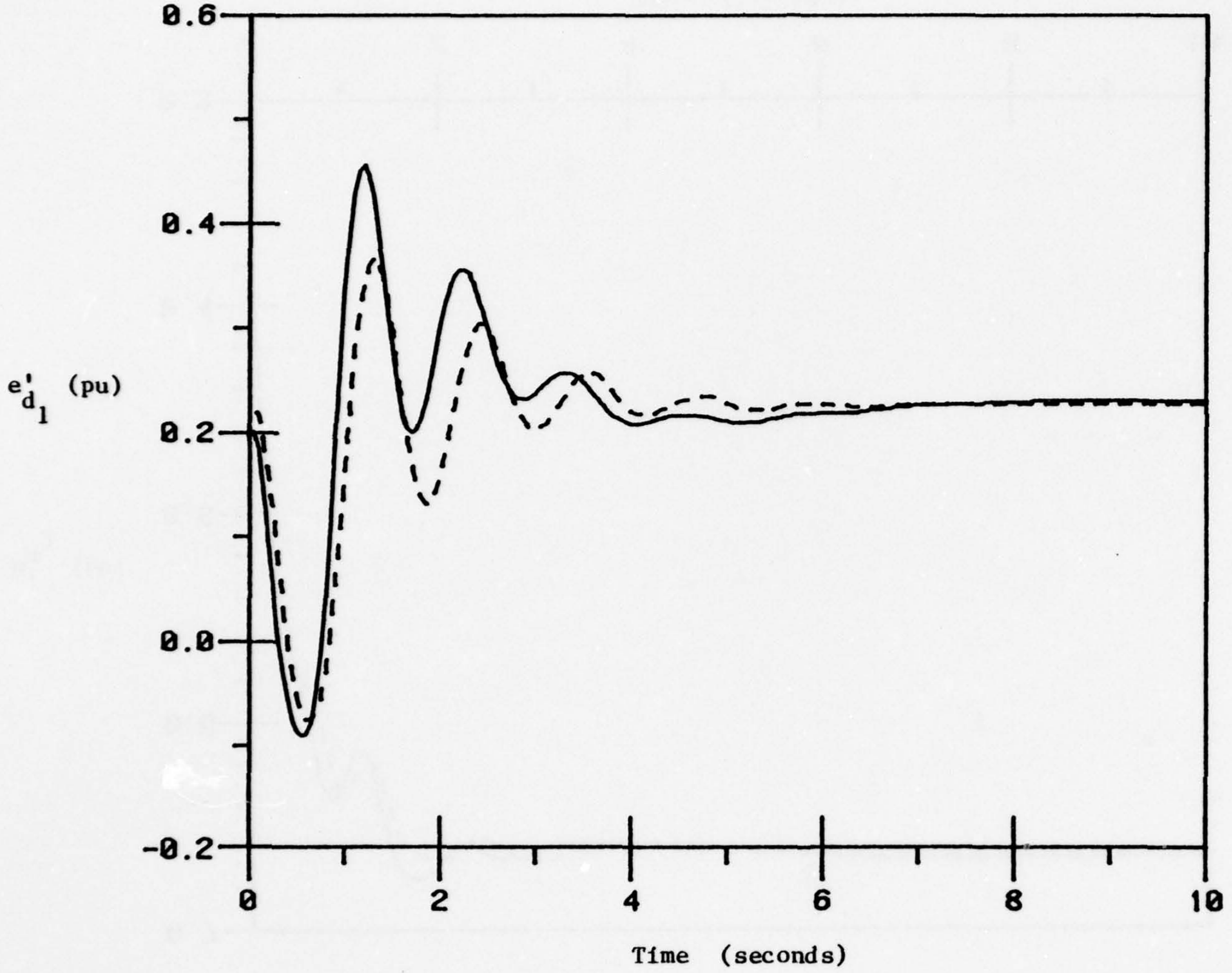


Fig. 4.30. Plots of exact and zero-order approximation of e'_{d_1}.

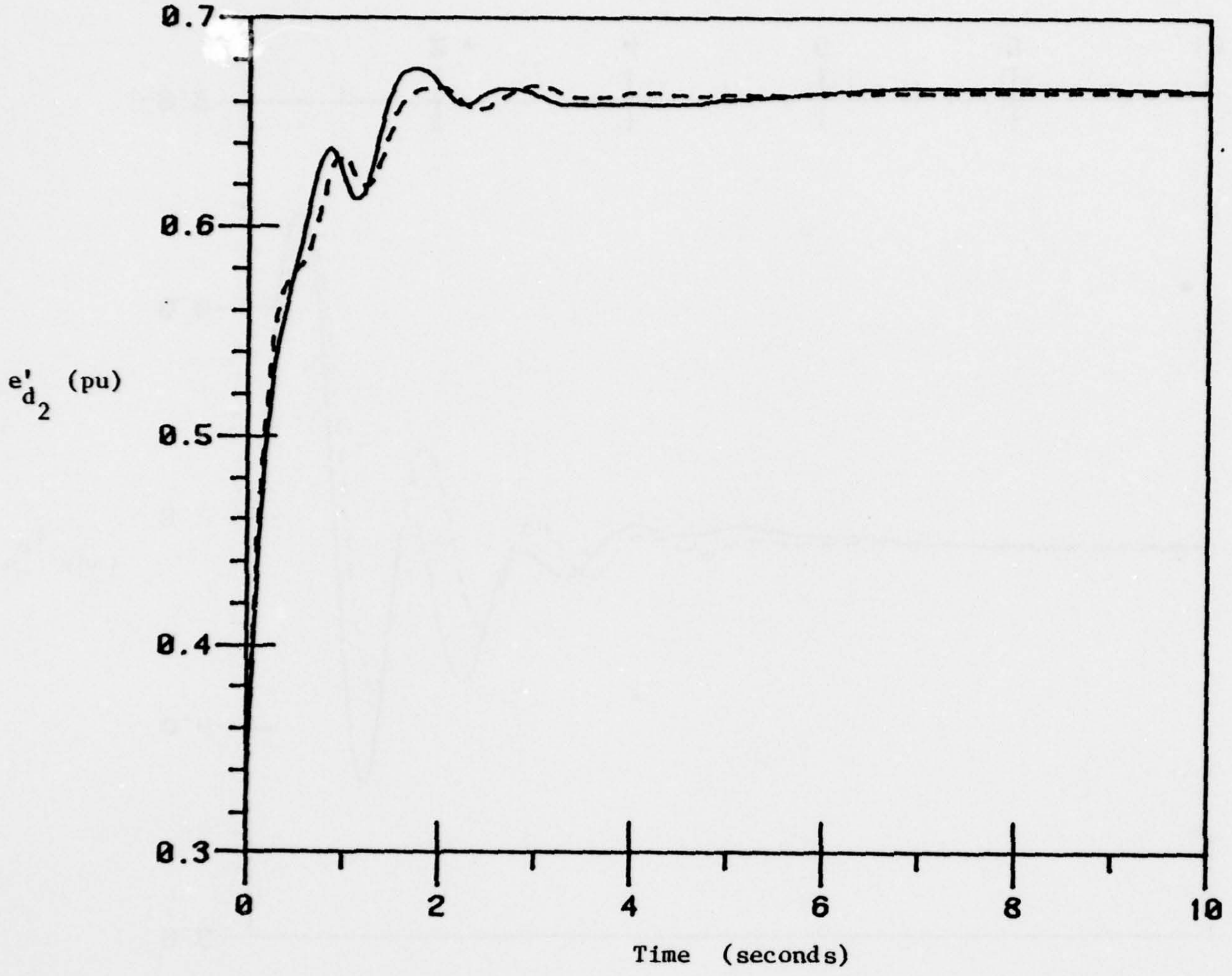


Fig. 4.31. Plots of exact and zero-order approximation of e'_{d_2}.

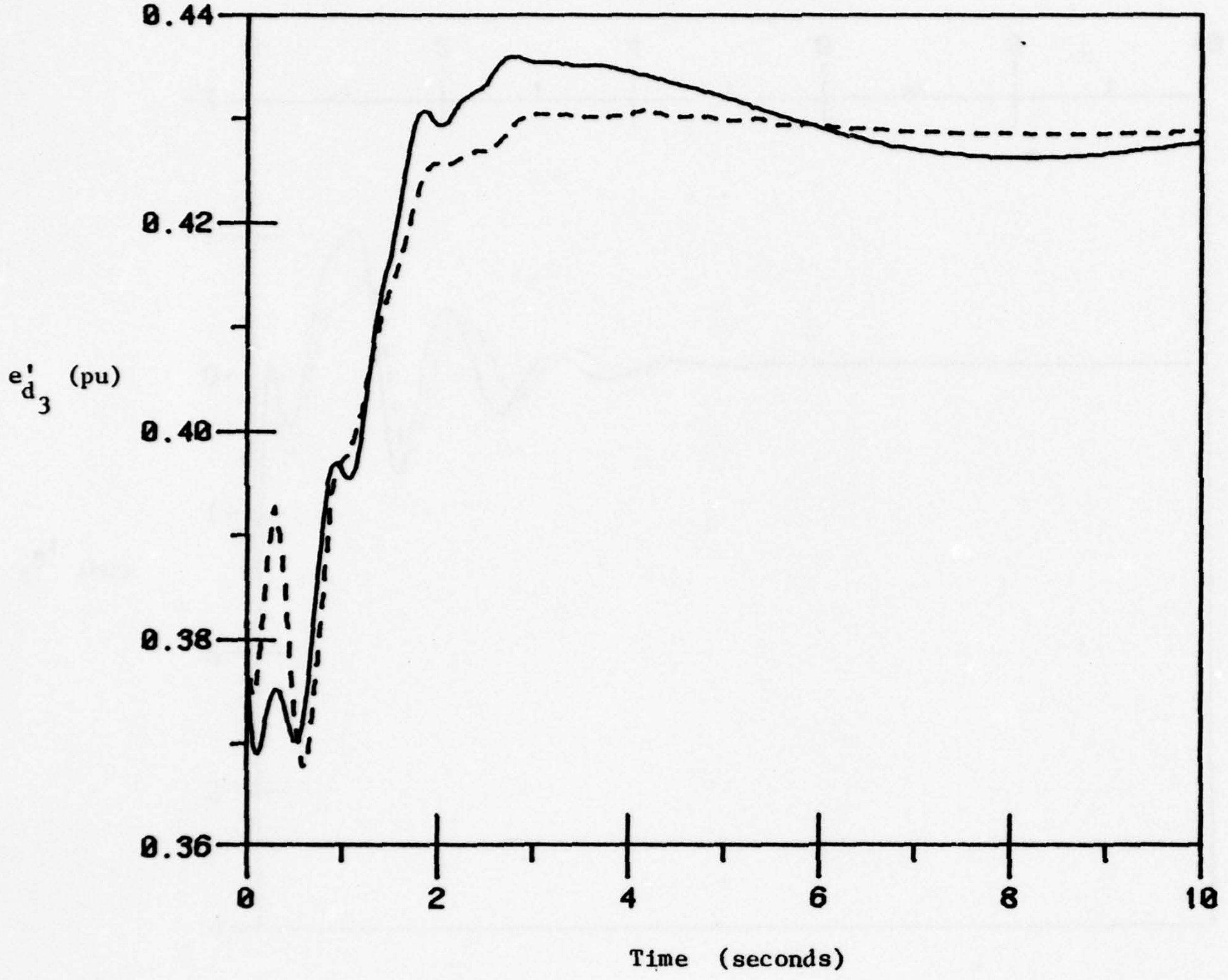


Fig. 4.32. Plots of exact and zero-order approximation of e'_{d_3}.

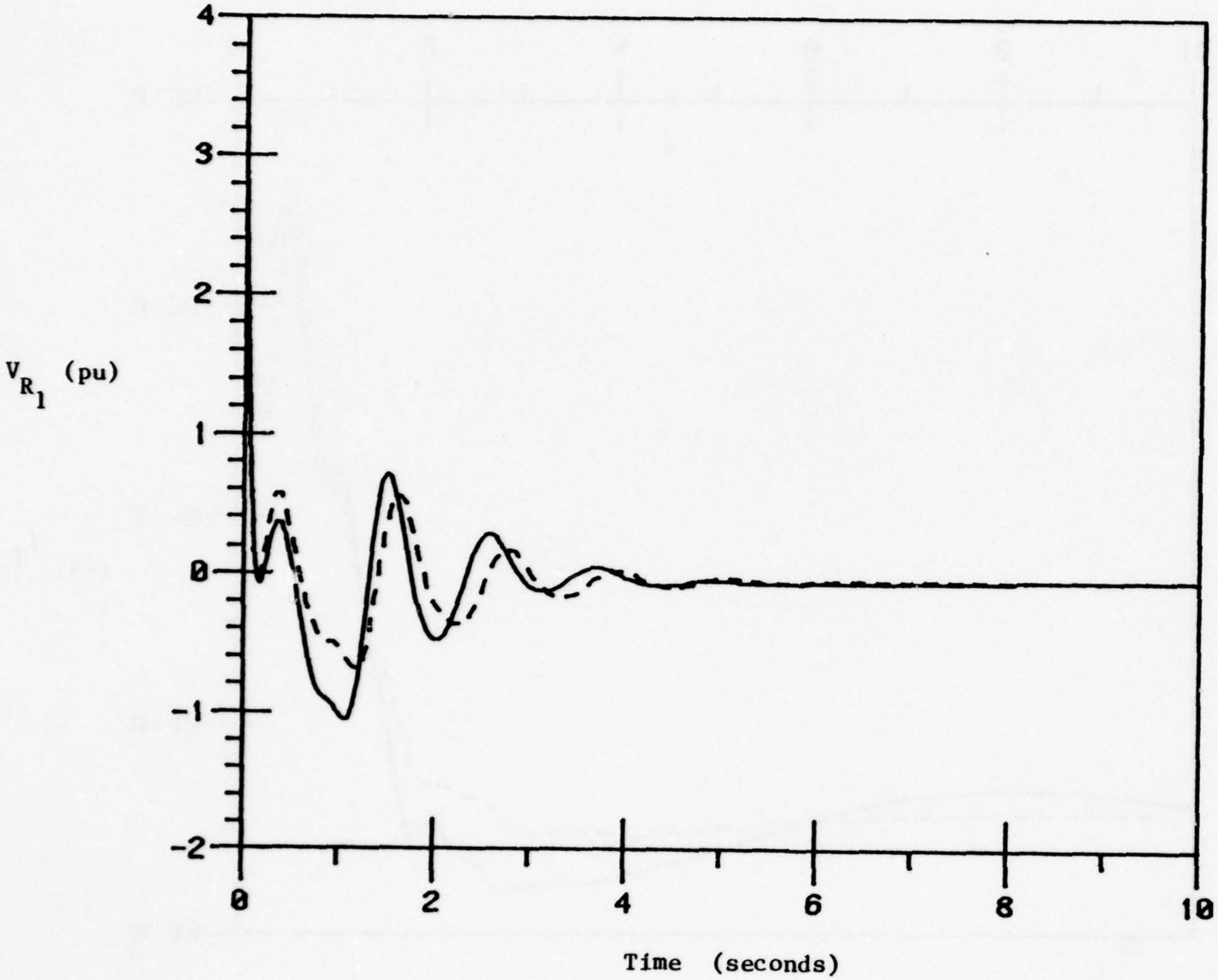


Fig. 4.33. Plots of exact and zero-order approximation of V_{R_1}.

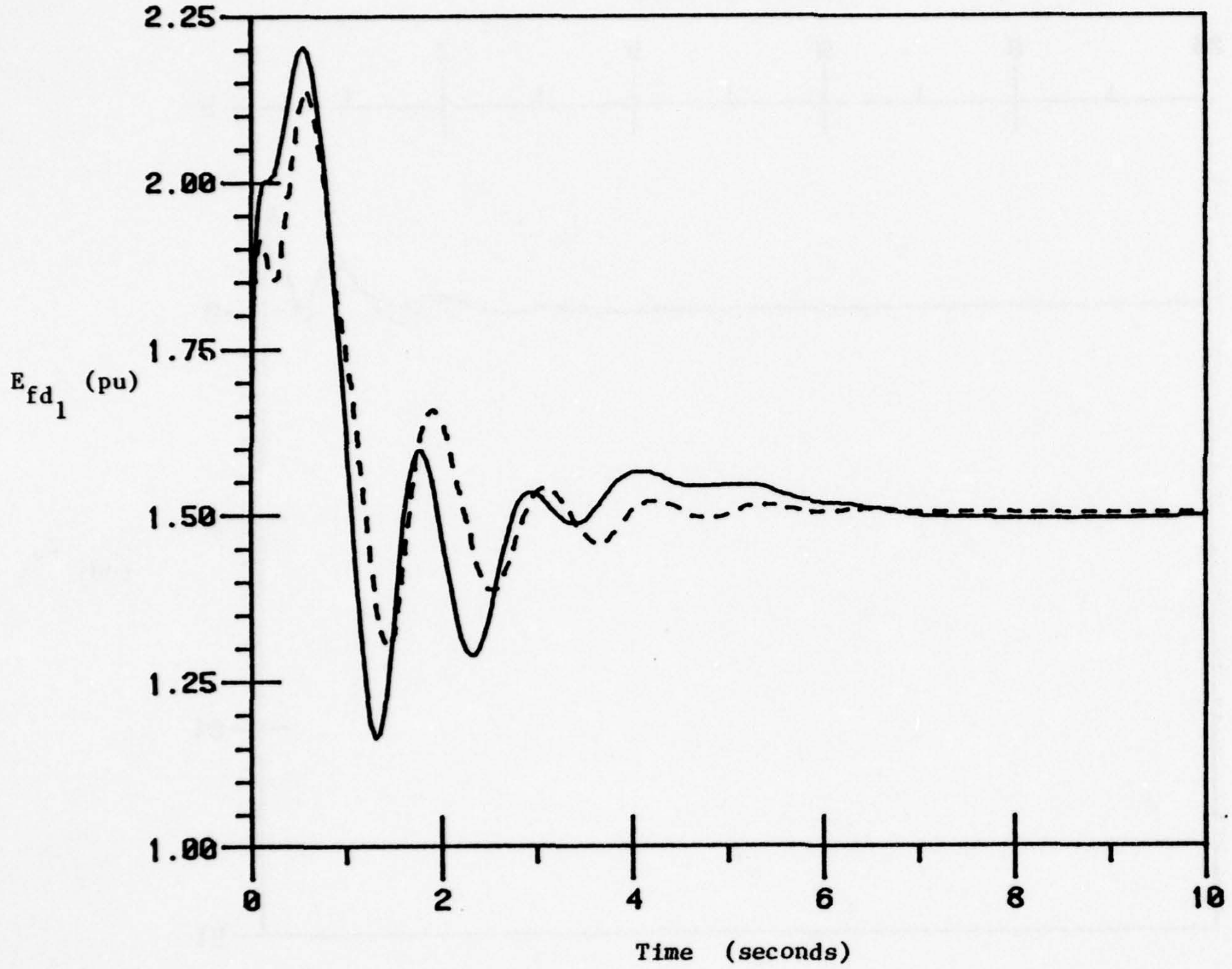


Fig. 4.34. Plots of exact and zero-order approximation of E_{fd_1}.

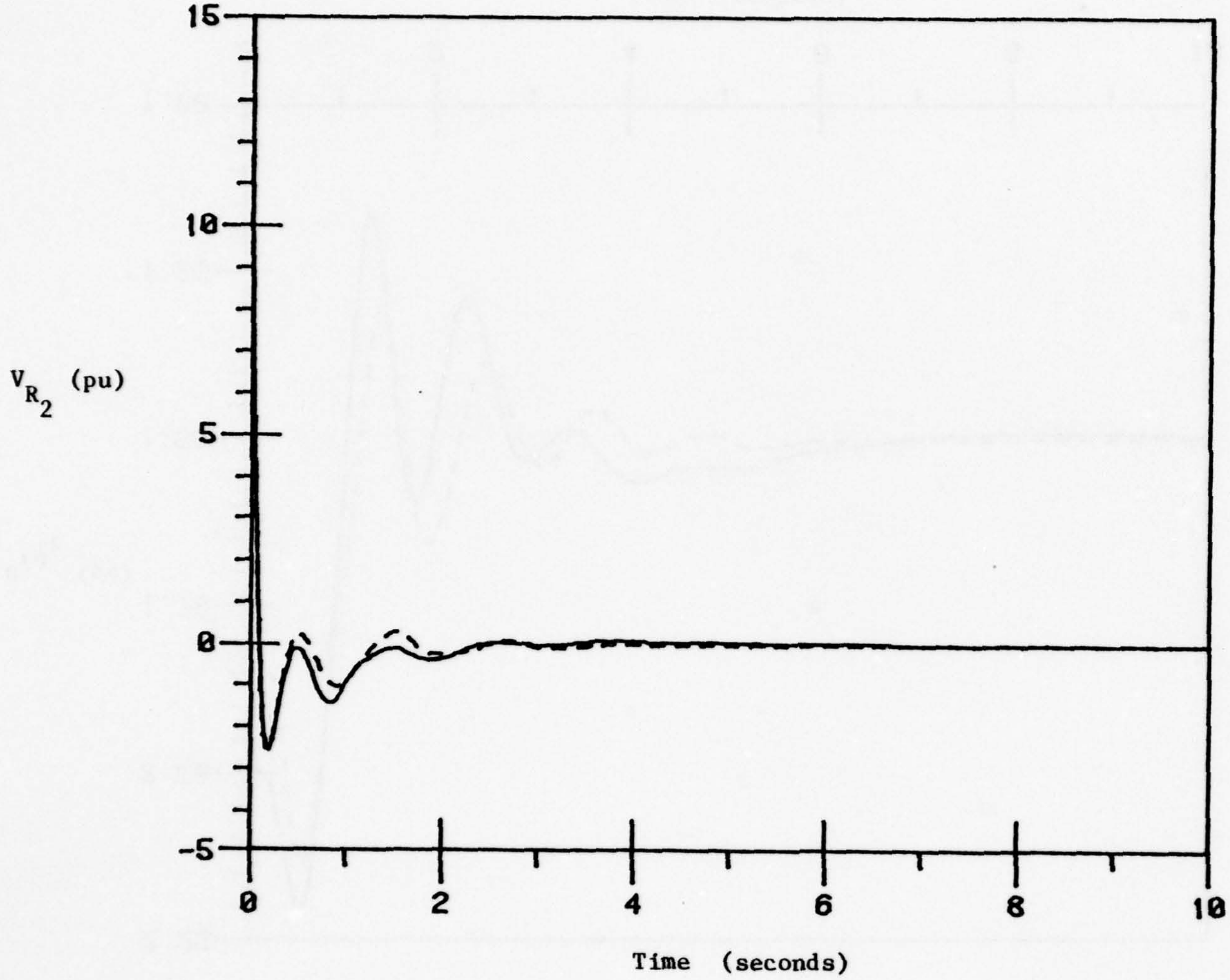


Fig. 4.35. Plots of exact and zero-order approximation of V_{R_2}.

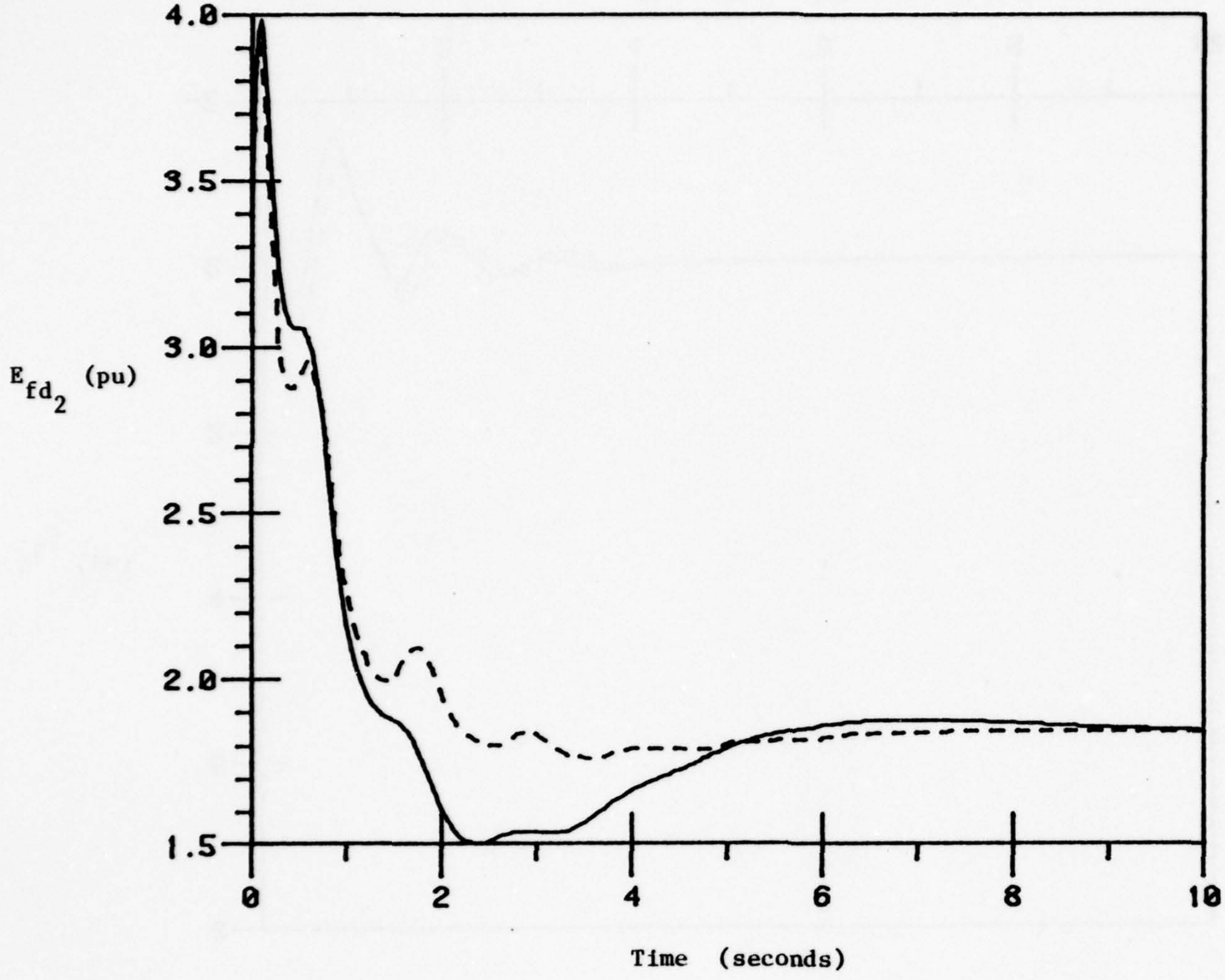


Fig. 4.36. Plots of exact and zero-order approximation of E_{fd_2}.

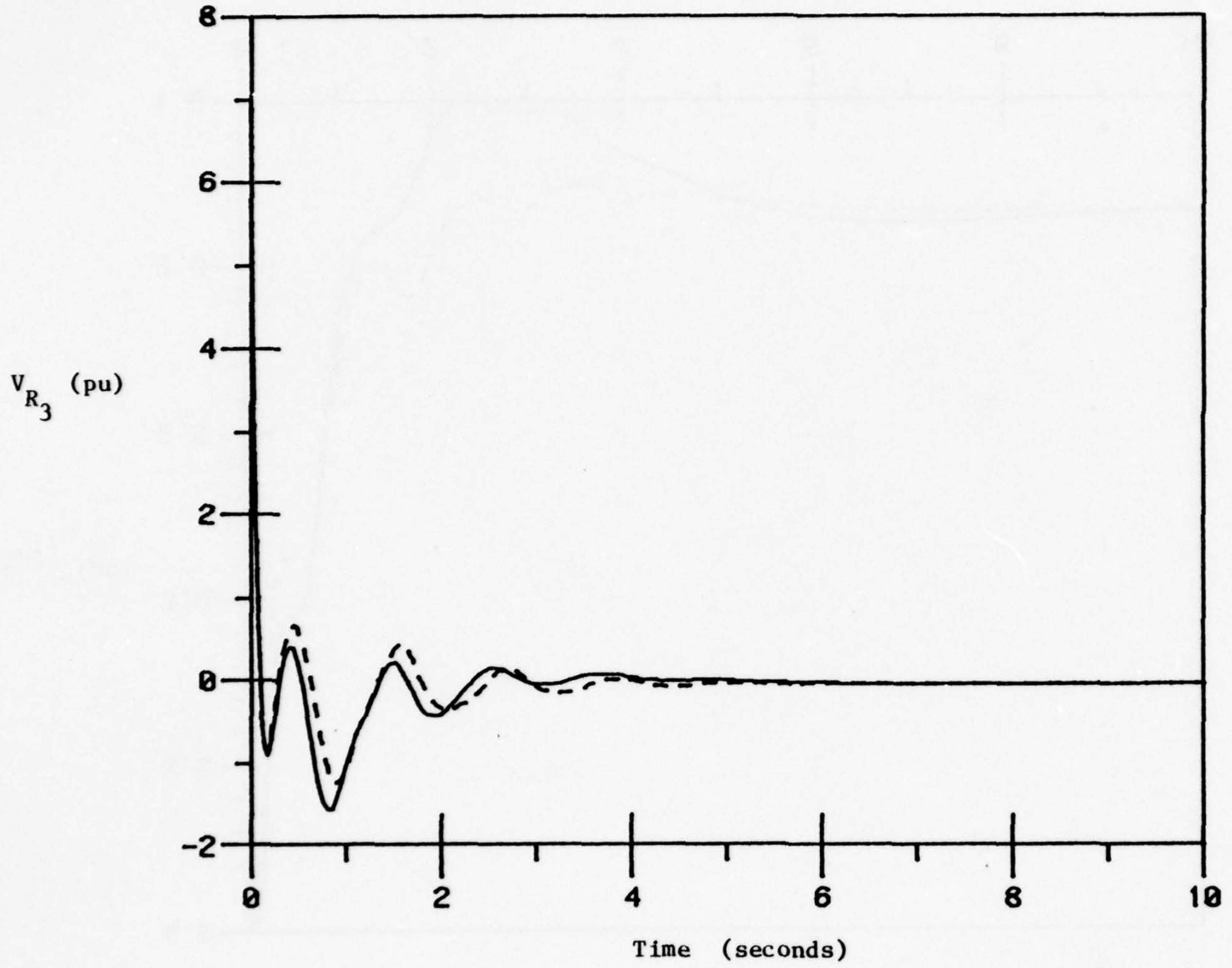


Fig. 4.37. Plots of exact and zero-order approximation of V_{R_3}.

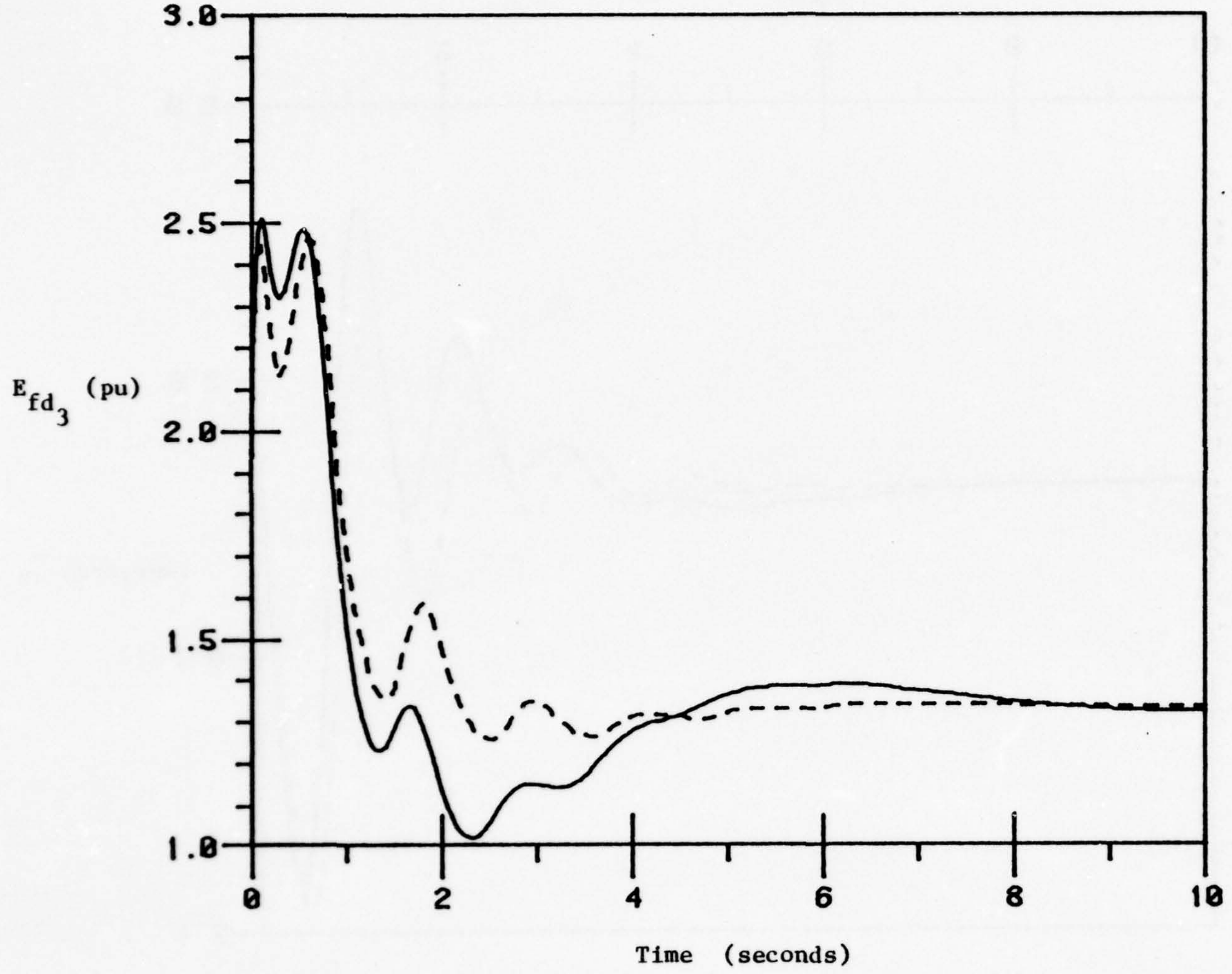


Fig. 4.38. Plots of exact and zero-order approximation of E_{fd_3}.

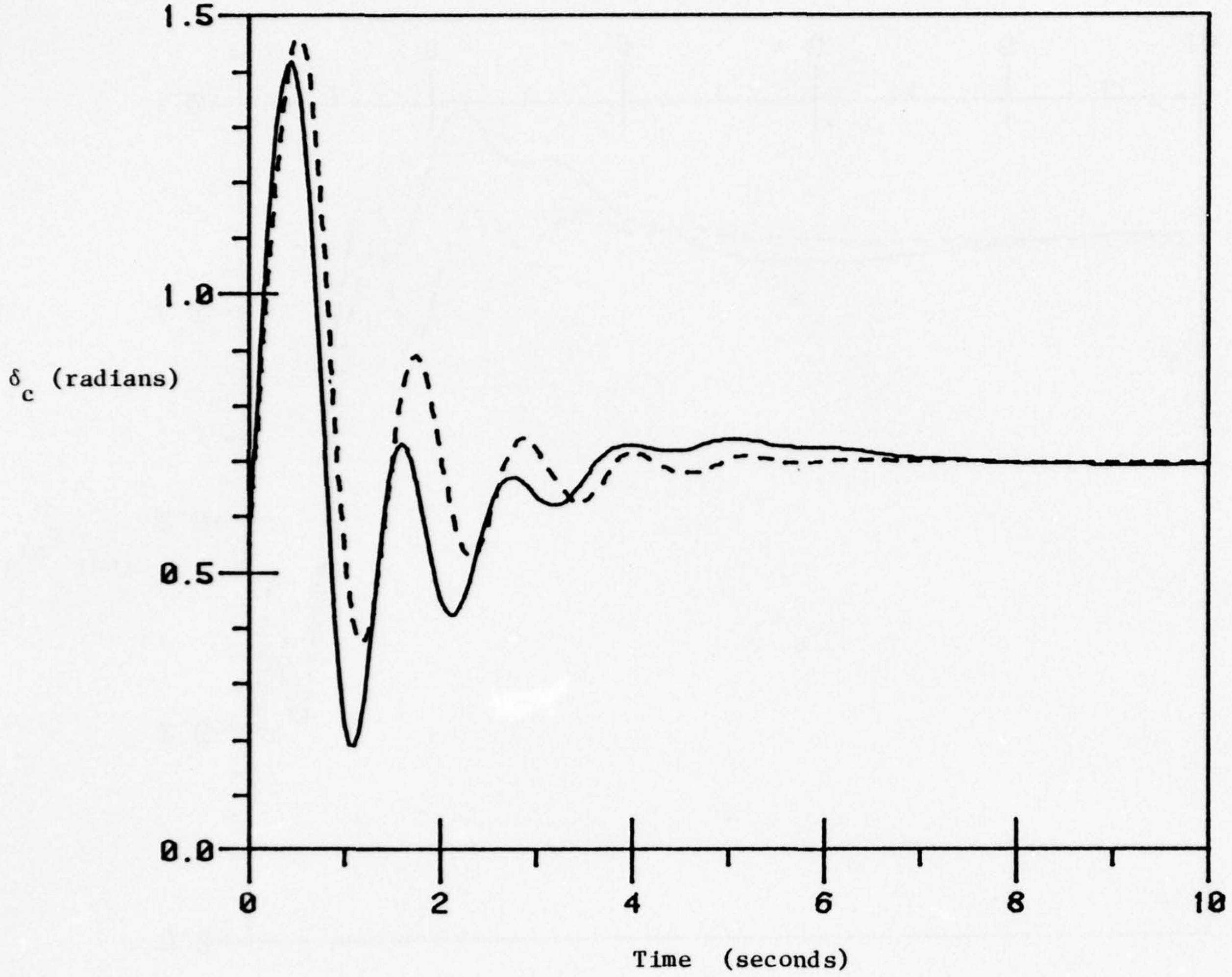


Fig. 4.39. Plots of exact and zero-order approximation of δ_c.

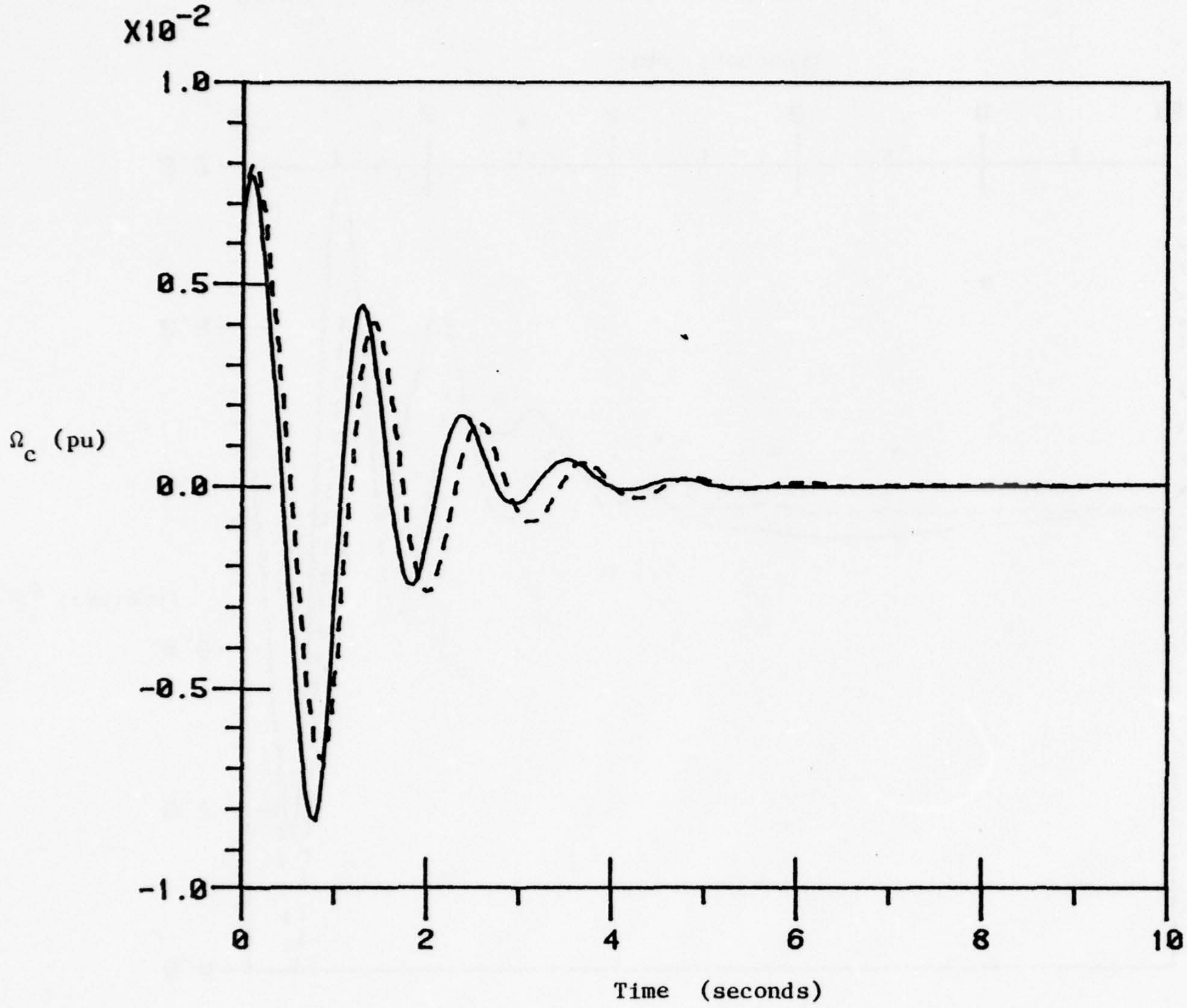


Fig. 4.40. Plots of exact and zero-order approximation of Ω_c.

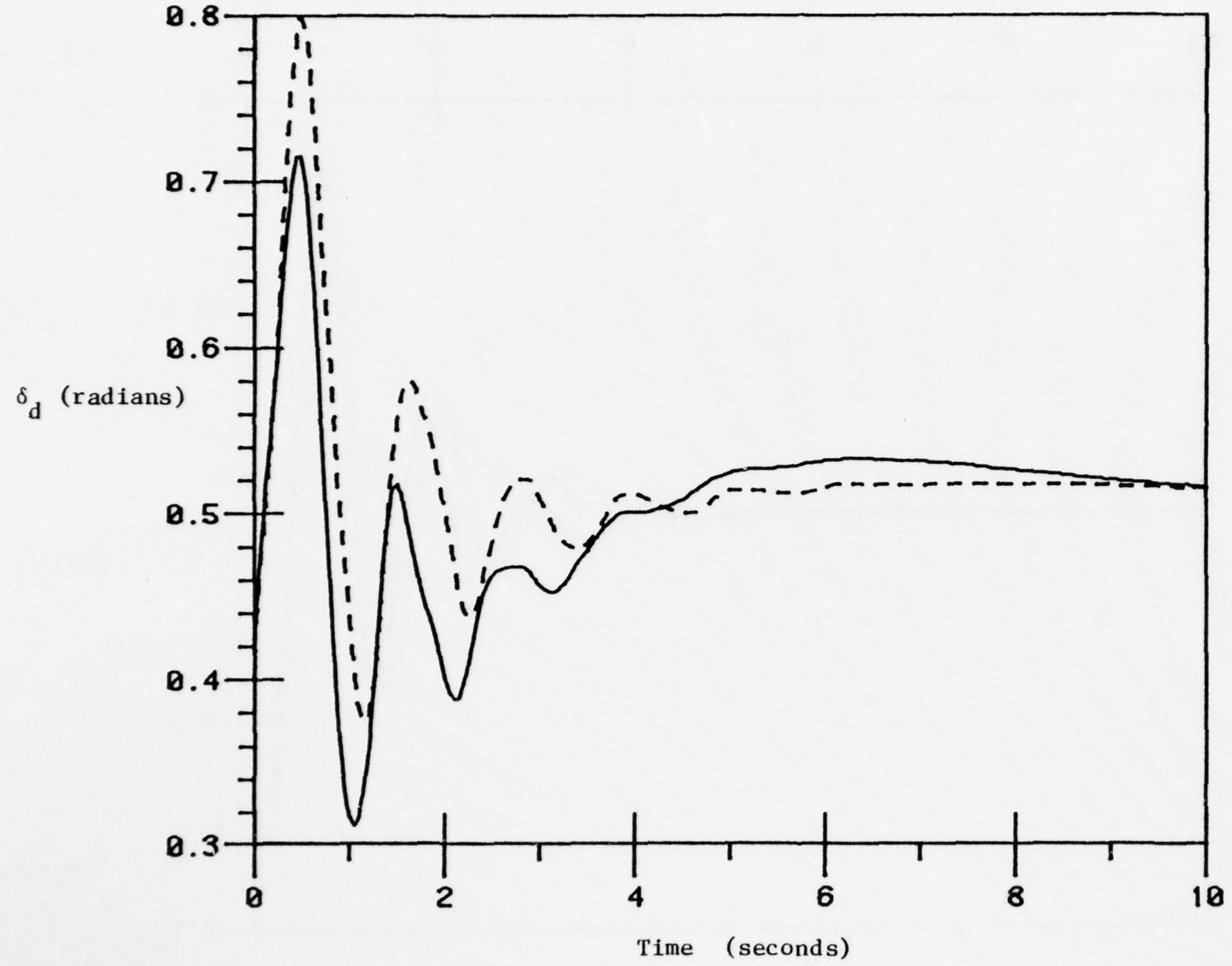


Fig. 4.41. Plots of exact and zero-order approximation of δ_d.

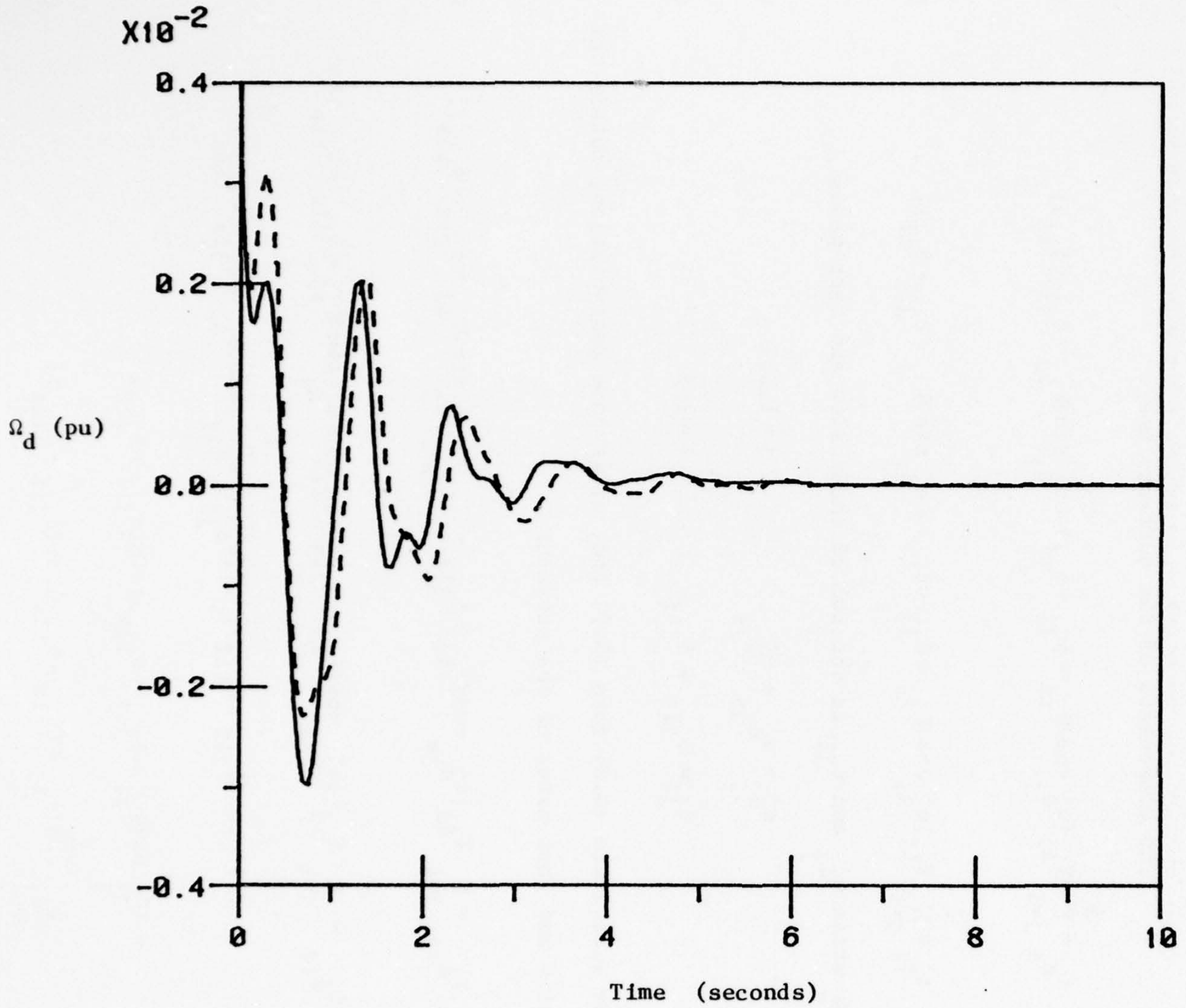


Fig. 4.42. Plots of exact and zero-order approximation of Ω_d.

equations is contributed by the machine currents. The expressions for the currents are given below in terms of the original angular variables. This is done for notational and computational simplicity. It is an easy matter to calculate the slow and fast parts of the original angles in terms of the respective parts of δ_c and δ_d since these two groups of angles are linearly related.

The components of the currents are

$$i_{d_i} = \sum_{j=1}^{3} Y_{ij}[e'_{d_j}\cos(\theta_{ij}+\delta_{j1}-\delta_{i1}) - e'_{q_j}\sin(\theta_{ij}+\delta_{j1}-\delta_{i1})] \tag{4.7a}$$

$$i_{q_i} = \sum_{j=1}^{3} Y_{ij}[e'_{q_j}\cos(\theta_{ij}+\delta_{j1}-\delta_{i1}) + e'_{d_j}\sin(\theta_{ij}+\delta_{j1}-\delta_{i1})] \quad . \tag{4.7b}$$

We write e'_{d_j} and δ_{j1} as the sum of their slow and fast parts

$$e'_{d_j} = e'_{d_{js}} + e'_{d_{jf}} \;, \qquad j = 1,2,3 \tag{4.8a}$$

$$\delta_{j1} = \delta_{j1s} + \delta_{j1f}, \qquad j = 2,3 \quad . \tag{4.8b}$$

We substitute (4.8) into (4.7) and, after some manipulation, obtain the slow and fast parts of the currents

$$i_{d_{is}} = \sum_{j=1}^{3} Y_{ij}[e'_{d_{js}}\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s}) - e'_{q_j}\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s})] \tag{4.9a}$$

$$\begin{aligned} i_{d_{if}} = \sum_{j=1}^{3} Y_{ij}\{&[e'_{d_{js}}\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s}) - e'_{q_j}\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s})] \times \\ &[-1+\cos(\delta_{j1f}-\delta_{i1f})] - [(e'_{d_{js}}+e'_{d_{jf}})\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s}) \\ &+ e'_{q_j}\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s})]\sin(\delta_{j1f}-\delta_{i1f}) \\ &+ e'_{d_{jf}}\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s})\cos(\delta_{j1f}-\delta_{i1f})\} \end{aligned} \tag{4.9b}$$

$$i_{q_{is}} = \sum_{j=1}^{3} Y_{ij}[e'_{q_j}\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s}) + e'_{d_{js}}\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s})] \tag{4.9c}$$

$$\begin{aligned} i_{q_{if}} = \sum_{j=1}^{3} Y_{ij}\{&[e'_{q_j}\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s}) \\ &+ e'_{d_{js}}\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s})][-1+\cos(\delta_{j1f}-\delta_{i1f})] \\ &+[(e'_{d_{js}}+e'_{d_{jf}})\cos(\theta_{ij}+\delta_{j1s}-\delta_{i1s})-e'_{q_j}\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s})]\sin(\delta_{j1f}-\delta_{i1f}) \\ &+ e'_{d_{jf}}\sin(\theta_{ij}+\delta_{j1s}-\delta_{i1s})\cos(\delta_{j1f}-\delta_{i1f})\} \ . \end{aligned} \tag{4.9d}$$

Approximation of Ω_r Equation

In order to apply the method of Chapter 3 to (4.5a), we first approximate the quotients as

$$\frac{P_{in_1}}{\Omega_r - h\Omega_c/H_1} \simeq \frac{P_{in_1}}{\Omega_r} + \frac{h}{H_1}\ \frac{P_{in_1}}{\Omega_r^{\ 2}}\,\Omega_c \tag{4.10a}$$

$$\frac{P_{in_2}}{\Omega_r+\Omega_c+H_3\Omega_d/h} \simeq \frac{P_{in_2}}{\Omega_r} - \frac{P_{in_2}}{\Omega_r^{\ 2}}\,(\Omega_c+H_3\Omega_d/h) \tag{4.10b}$$

$$\frac{P_{in_3}}{\Omega_r+\Omega_c-H_2\Omega_d/h} \simeq \frac{P_{in_3}}{\Omega_r} - \frac{P_{in_3}}{\Omega_r^{\ 2}}\,(\Omega_c-H_2\Omega_d/h) \ . \tag{4.10c}$$

Equation (4.10) follows since, generally, Ω_c and Ω_d are small. Now we write e'_{d_i}, i_{d_i}, and i_{q_i} as the sum of their slow and fast parts. The sums in (4.5a) can be split as follows

$$\text{slow:} \quad \sum_{i=1}^{3} (e'_{q_i} i_{q_{is}} + e'_{d_{is}} i_{d_{is}}) \tag{4.11a}$$

$$\text{fast:} \quad \sum_{i=1}^{3} [e'_{q_i} i_{q_{if}} + (e'_{d_{is}} + e'_{d_{if}}) i_{d_{if}} + e'_{d_{if}} i_{d_{is}}] \; . \tag{4.11b}$$

Finally, since $\Omega_{cs} = \Omega_{ds} = 0$, the slow part of Ω_r is approximated by the solution of

$$\dot{\Omega}_{rs} = \frac{1}{2H} [\frac{P_{in_1} + P_{in_2} + P_{in_3}}{\Omega_{rs}} - D(\Omega_{rs} - 1) - \sum_{i=1}^{3} (e'_{q_{is}} i_{q_{is}} + e'_{d_{is}} i_{d_{is}})],$$

$$\Omega_{rs}(0) = \Omega_r(0) + c_1 + c_2 + c_3 + c_4 + c_5 + c_6 + c_7 \tag{4.12}$$

and the fast part is approximated as

$$\Omega_{rf} = -c_1 + \frac{h}{H_1} \frac{P_{in_1}}{2H} \int_0^t \frac{\Omega^o_{cf}}{\Omega^{o2}_r} d\tau - c_2 - \frac{P_{in_2}}{2H} \int_0^t (\frac{\Omega^o_{cf} + H_3 \Omega^o_{df}/h}{\Omega^{o2}_r}) d\tau$$

$$-c_3 - \frac{P_{in_3}}{2H} \int_0^t (\frac{\Omega^o_{cf} - H_2 \Omega^o_{df}/h}{\Omega^{o2}_r}) d\tau - c_4 + \frac{D_1}{H_1} \frac{h}{2H} \int_0^t \Omega^o_{cf} d\tau$$

$$-c_5 - \frac{D_2}{2H} \int_0^t (\Omega^o_{cf} + H_3 \Omega^o_{df}/h) d\tau - c_6 - \frac{D_3}{2H} \int_0^t (\Omega^o_{cf} - H_2 \Omega^o_{df}/h) d\tau$$

$$-c_7 - \frac{1}{2H} \int_0^t \{ \sum_{i=1}^{3} [e'^o_{q_i} i^o_{q_{if}} + (e'^o_{d_{is}} + e'^o_{d_{if}}) i^o_{d_{if}} + e'^o_{d_{if}} i^o_{d_{is}}]\} d\tau \tag{4.13}$$

where

$$c_1 = \frac{h}{H_1} \frac{P_{in_1}}{2H} \int_0^\infty \frac{\Omega^o_{cf}}{\Omega^{o2}_r} d\tau \tag{4.14a}$$

$$c_2 = - \frac{P_{in_2}}{2H} \int_0^\infty (\frac{\Omega^o_{cf} + H_3 \Omega^o_{df}/h}{\Omega^{o2}_r}) d\tau \tag{4.14b}$$

$$c_3 = - \frac{P_{in_3}}{2H} \int_0^{\infty} \left(\frac{\Omega^o_{cf} - H_2\Omega^o_{df}/h}{\Omega_r^{o2}}\right) d\tau \tag{4.14c}$$

$$c_4 = \frac{D_1}{H_1} \; \frac{h}{2H} \int_0^{\infty} \Omega^o_{cf} \; d\tau \tag{4.14d}$$

$$c_5 = - \frac{D_2}{2H} \int_0^{\infty} (\Omega^o_{cf} + H_3\Omega^o_{df}/h) \; d\tau \tag{4.14e}$$

$$c_6 = - \frac{D_3}{2H} \int_0^{\infty} (\Omega^o_{cf} - H_2\Omega^o_{df}/h) \; d\tau \tag{4.14f}$$

$$c_7 = - \frac{1}{2H} \int_0^{\infty} \{ \sum_{i=1}^{3} \; [e'^o_{q_i} \; i^o_{q_{if}} + (e'^o_{d_{is}} + e'^o_{d_{if}}) i^o_{d_{if}} + e'^o_{d_{if}} \; i^o_{d_{is}}]\} \; d\tau \; . \tag{4.14g}$$

Approximation of e'_q Equations

In (4.5c) we write i_{d_i} and E_{fd_i} as the sum of their slow and fast parts. The slow part of e'_{q_i} is given approximately by the solution of

$$\dot{e}'_{q_{is}} = \frac{1}{T'_{d_{0i}}} [- e'_{q_{is}} - (x_{d_i} - x'_i) i_{d_{is}} + E_{fd_{is}}],$$

$$e'_{q_{is}}(0) = e'_{q_i}(0) + c_8 + c_9 \tag{4.15}$$

and the fast part is approximated by

$$e'_{q_{if}} = -c_8 - \frac{(x_{d_i} - x'_i)}{T'_{d_{0i}}} \int_0^t i^o_{d_{if}} d\tau - c_9 + \frac{1}{T'_{d_{0i}}} \int_0^t E^o_{fd_{if}} d\tau \tag{4.16}$$

where

$$c_8 = - \frac{(x_{d_i} - x_i')}{T'_{d_{0i}}} \int_0^\infty i^o_{d_{if}} d\tau \qquad (4.17a)$$

$$c_9 = \frac{1}{T'_{d_{0i}}} \int_0^\infty E^o_{fd_{if}} d\tau \quad . \qquad (4.17b)$$

Approximation of R_f Equations

Writing E_{fd_i} as the sum of its slow and fast parts in (4.5b) gives the following for the approximation of the slow part of R_{f_i}

$$\dot{R}_{f_{is}} = \frac{1}{T_{F_i}} (- R_{f_{is}} + \frac{K_{F_i}}{T_{F_i}} E_{fd_{is}}) \ ,$$

$$R_{f_{is}}(0) = R_{f_i}(0) + c_{10}. \qquad (4.18)$$

The fast part is approximated by

$$R_{f_{if}} = -c_{10} + \frac{K_{F_i}}{T_{F_i}^2} \int_0^t E^o_{fd_{if}} d\tau \qquad (4.19)$$

where

$$c_{10} = \frac{K_{F_i}}{T_{F_i}^2} \int_0^\infty E^o_{fd_{if}} d\tau \quad . \qquad (4.20)$$

The results of the calculations are shown in Figs. 4.43 - 4.62. These plots were made after 3 passes through the iterative cycle. Regarding the slow variables (Figs. 4.43-4.49), it is clear that a fast component has been added, but the "amount" of fast component is not enough.

The fast variables (Figs. 4.50 - 4.62) also show some improvement over their zero-order counterparts. In particular, the "frequency" of the slower rotor angle oscillatory mode is more accurate in the corrected case. The increased accuracy of this mode favorably affects many of the other fast

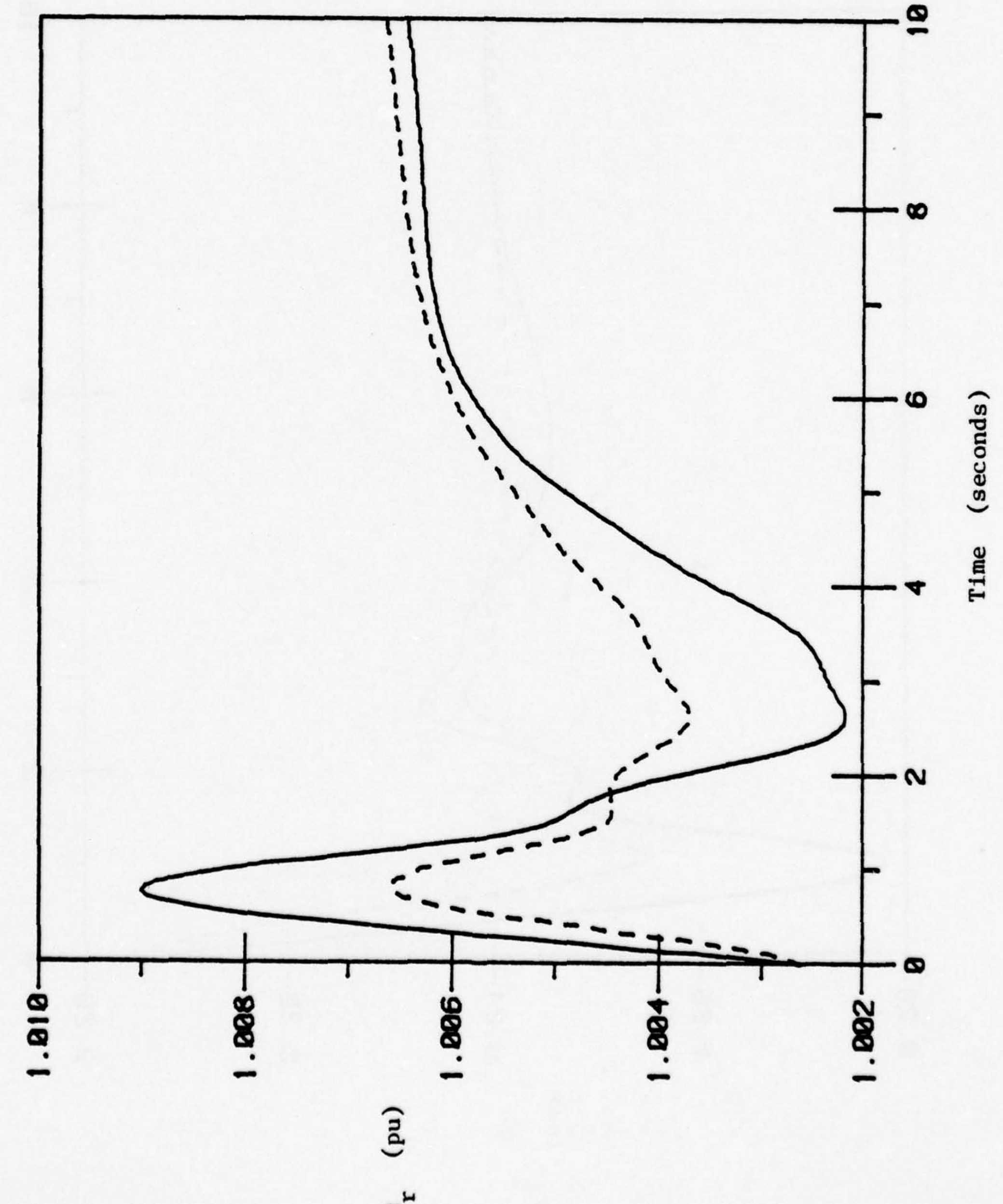


Fig. 4.43. Plots of exact and corrected approximation of Ω_r.

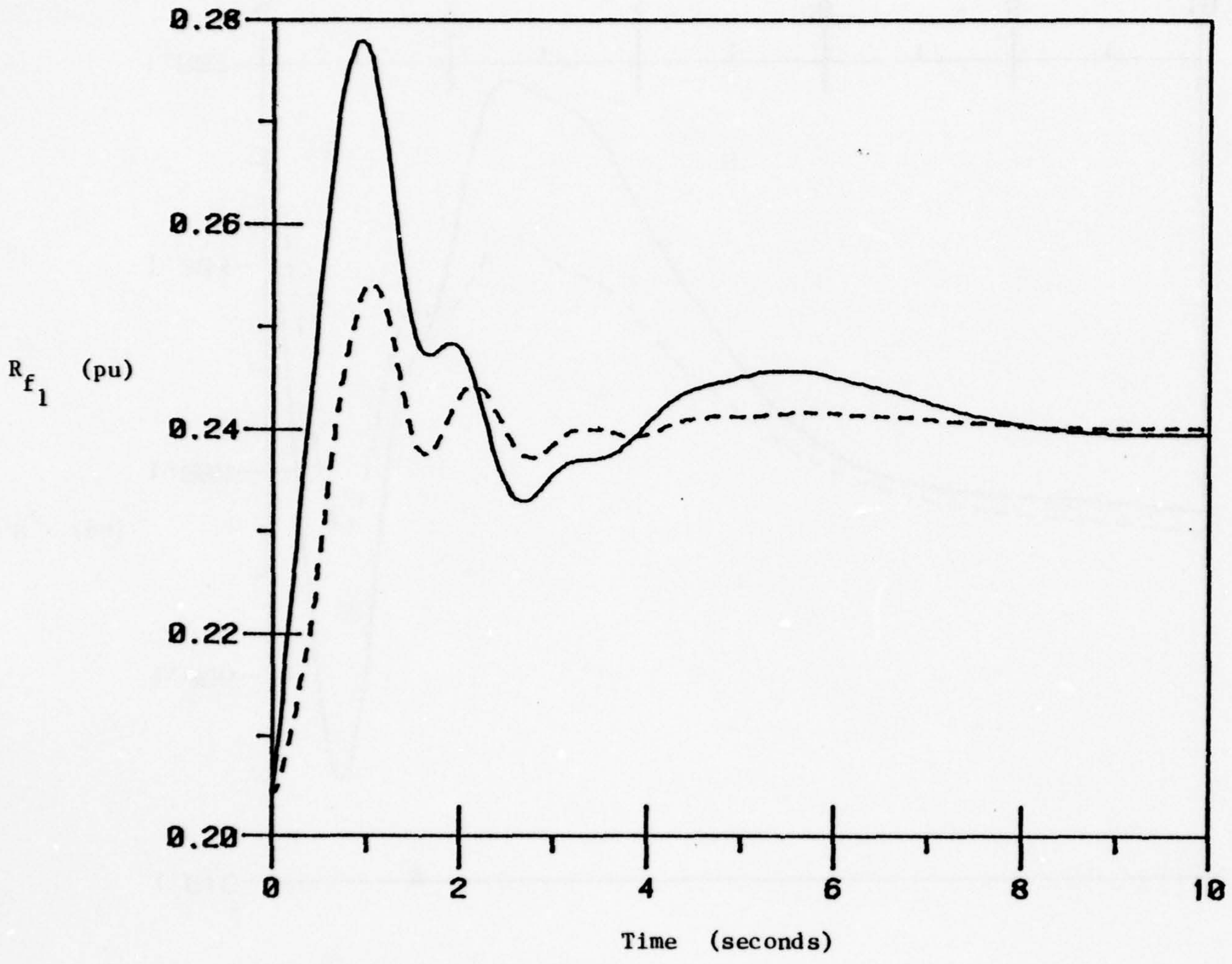


Fig. 4.44. Plots of exact and corrected approximation of R_{f_1}.

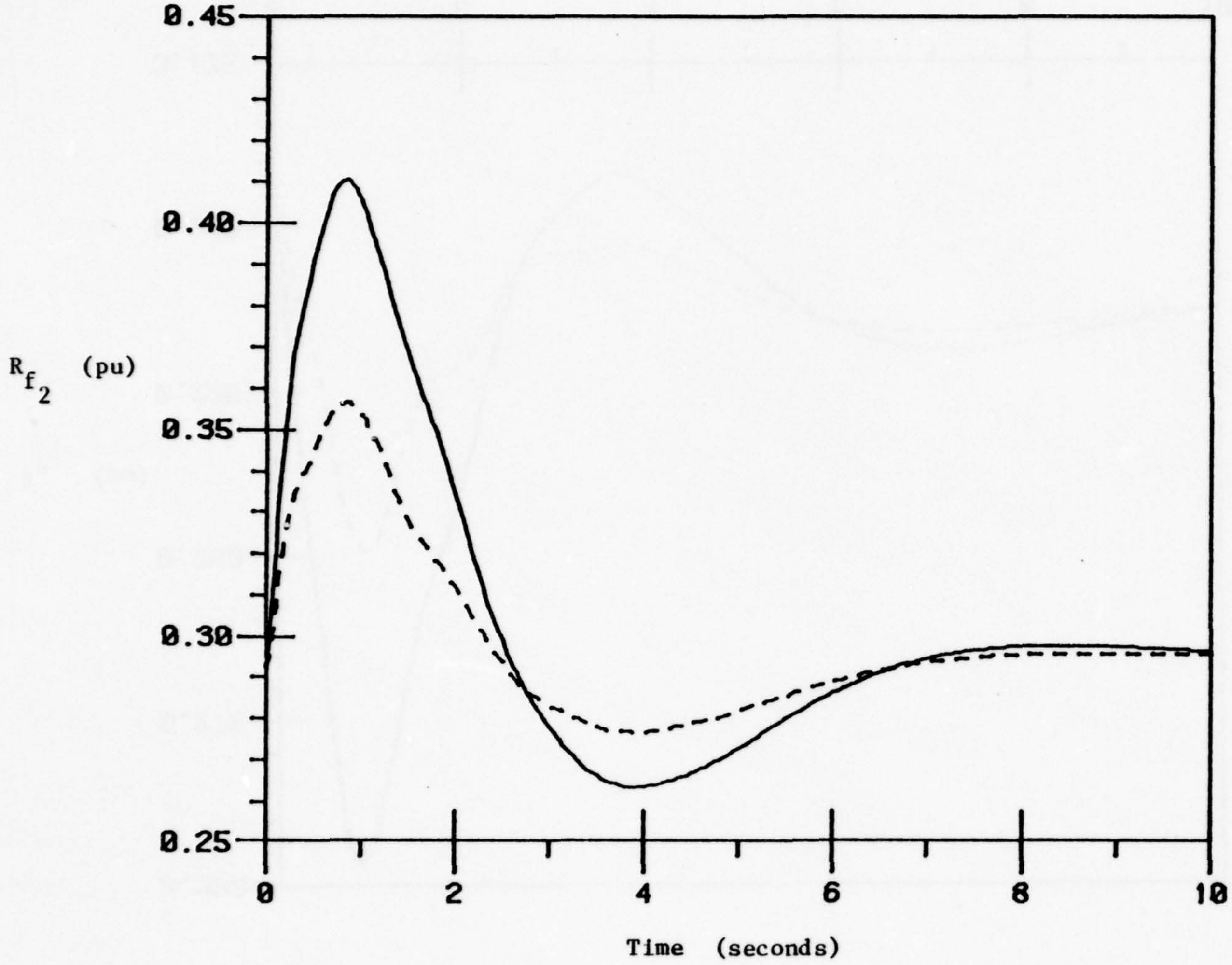


Fig. 4.45. Plots of exact and corrected approximation of R_{f_2}.

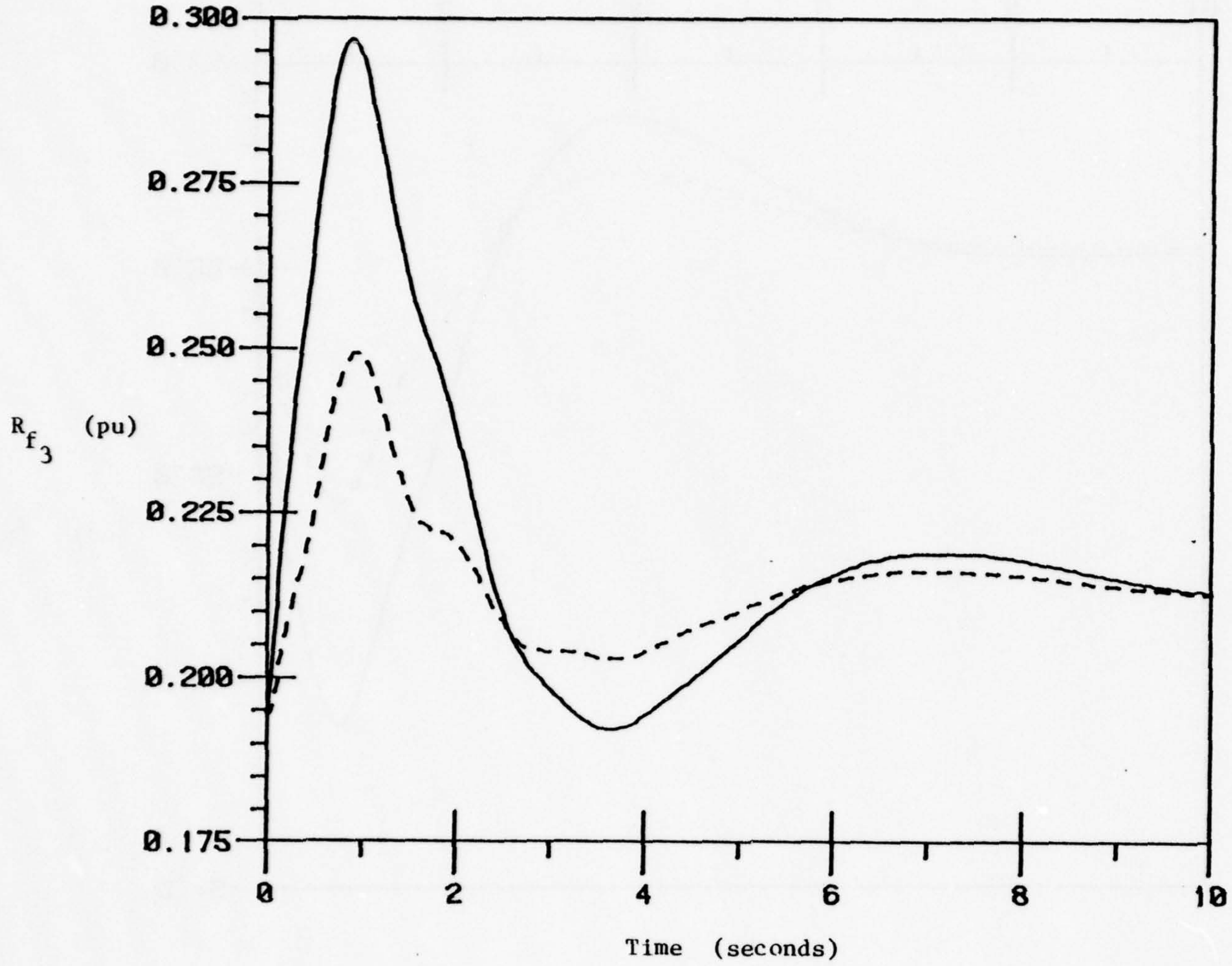


Fig. 4.46. Plots of exact and corrected approximation of R_{f_3}.

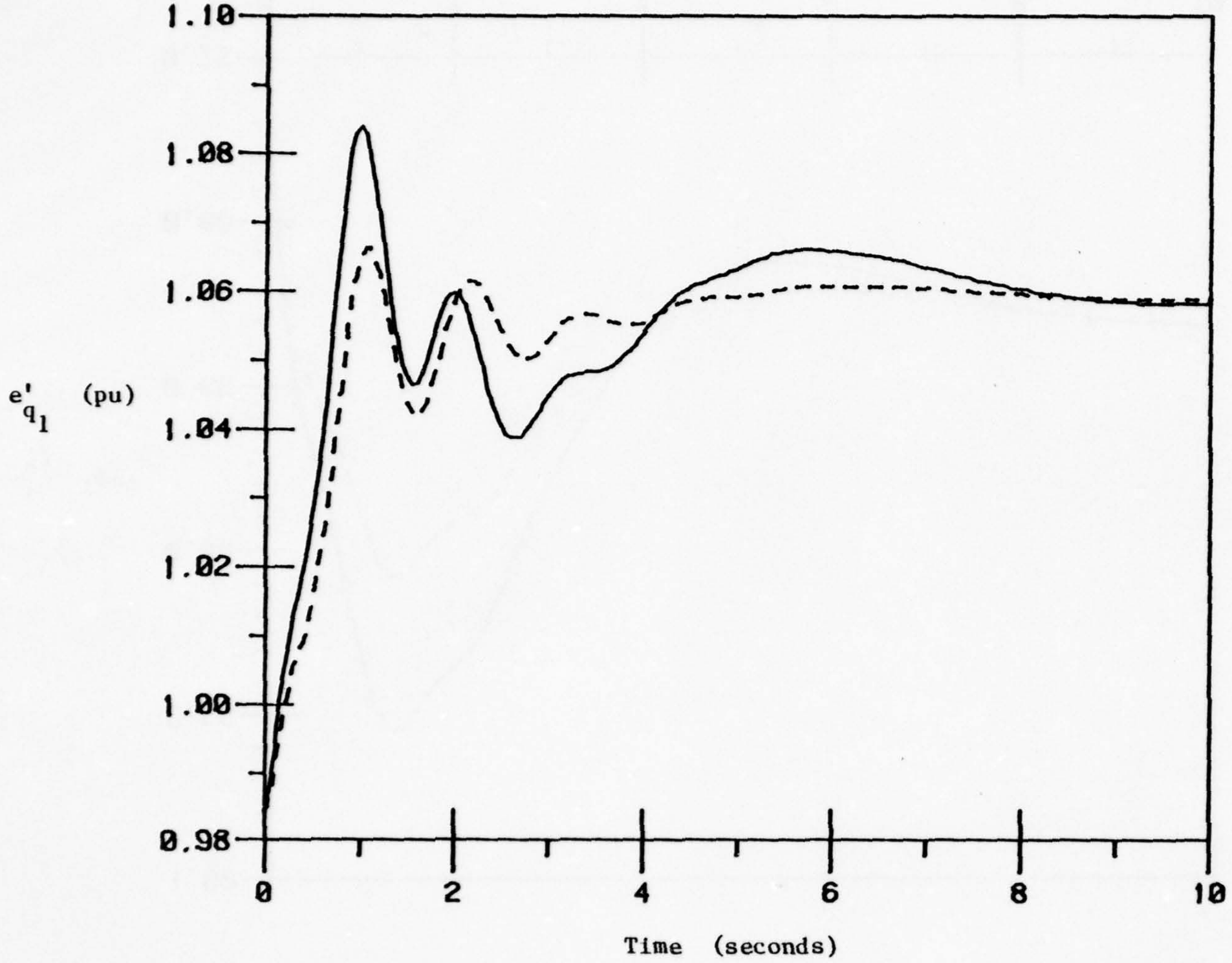


Fig. 4.47. Plots of exact and corrected approximation of e'_{q_1}.

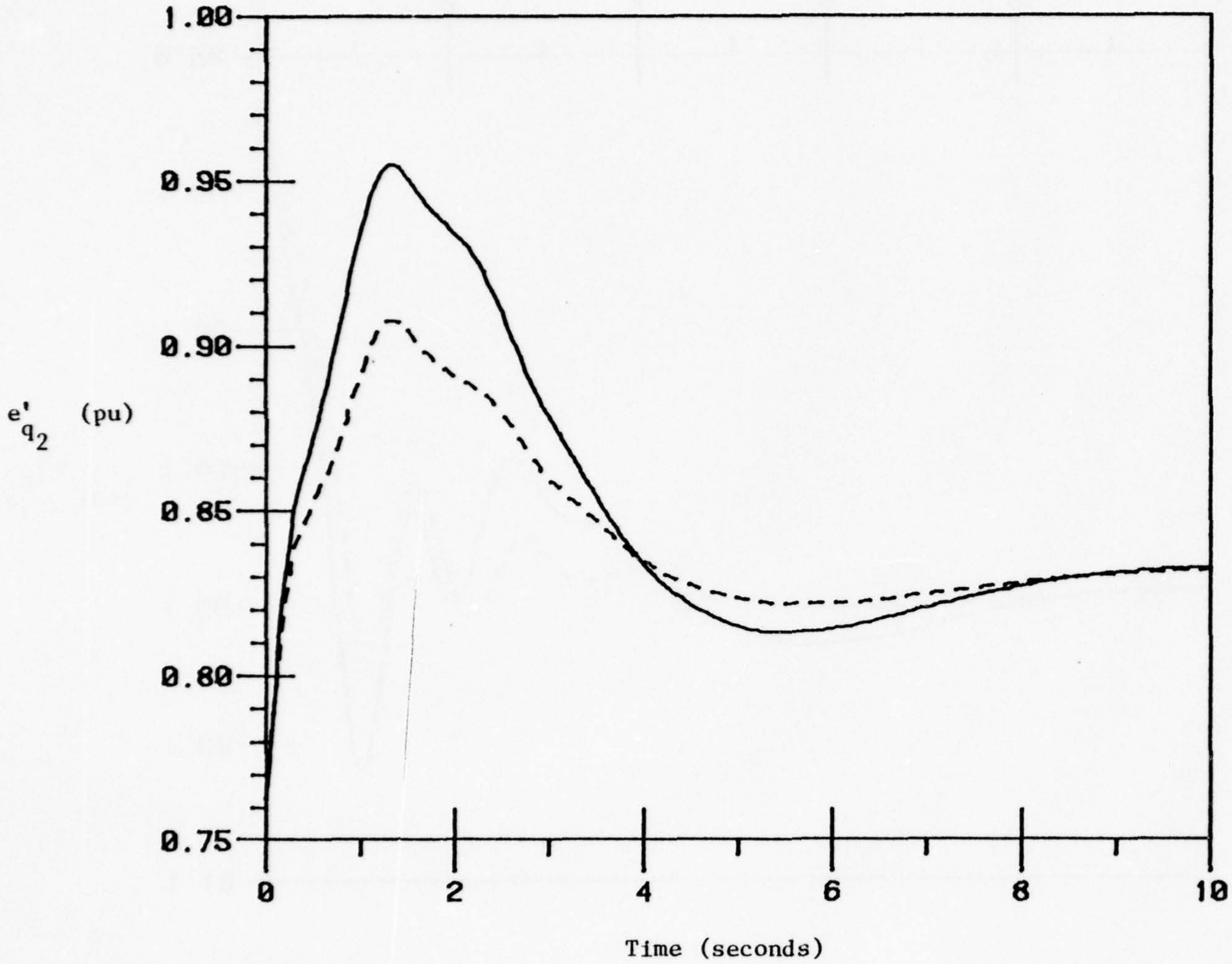


Fig. 4.48. Plots of exact and corrected approximation of e'_{q_2}.

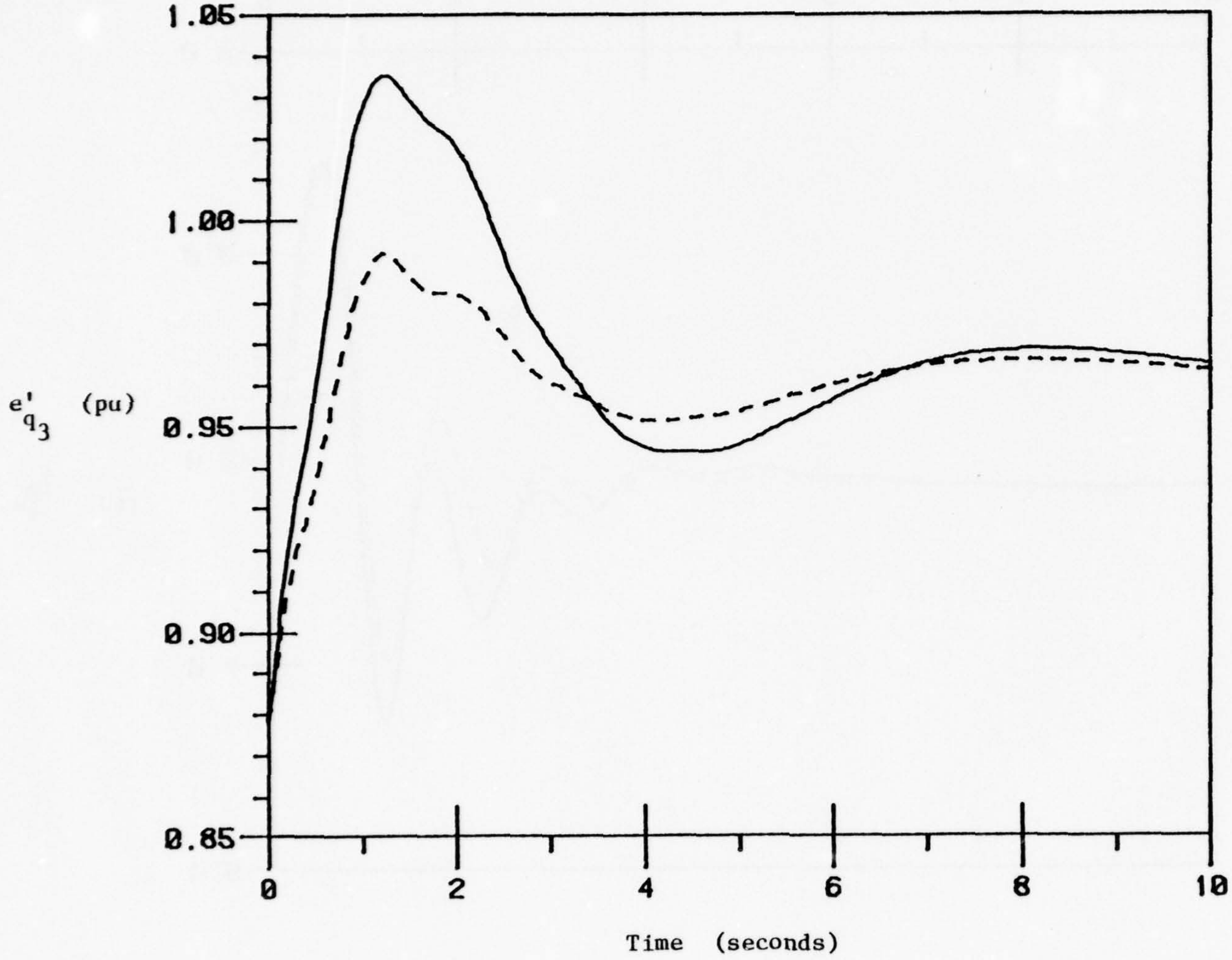


Fig. 4.49. Plots of exact and corrected approximation of e'_{q_3}.

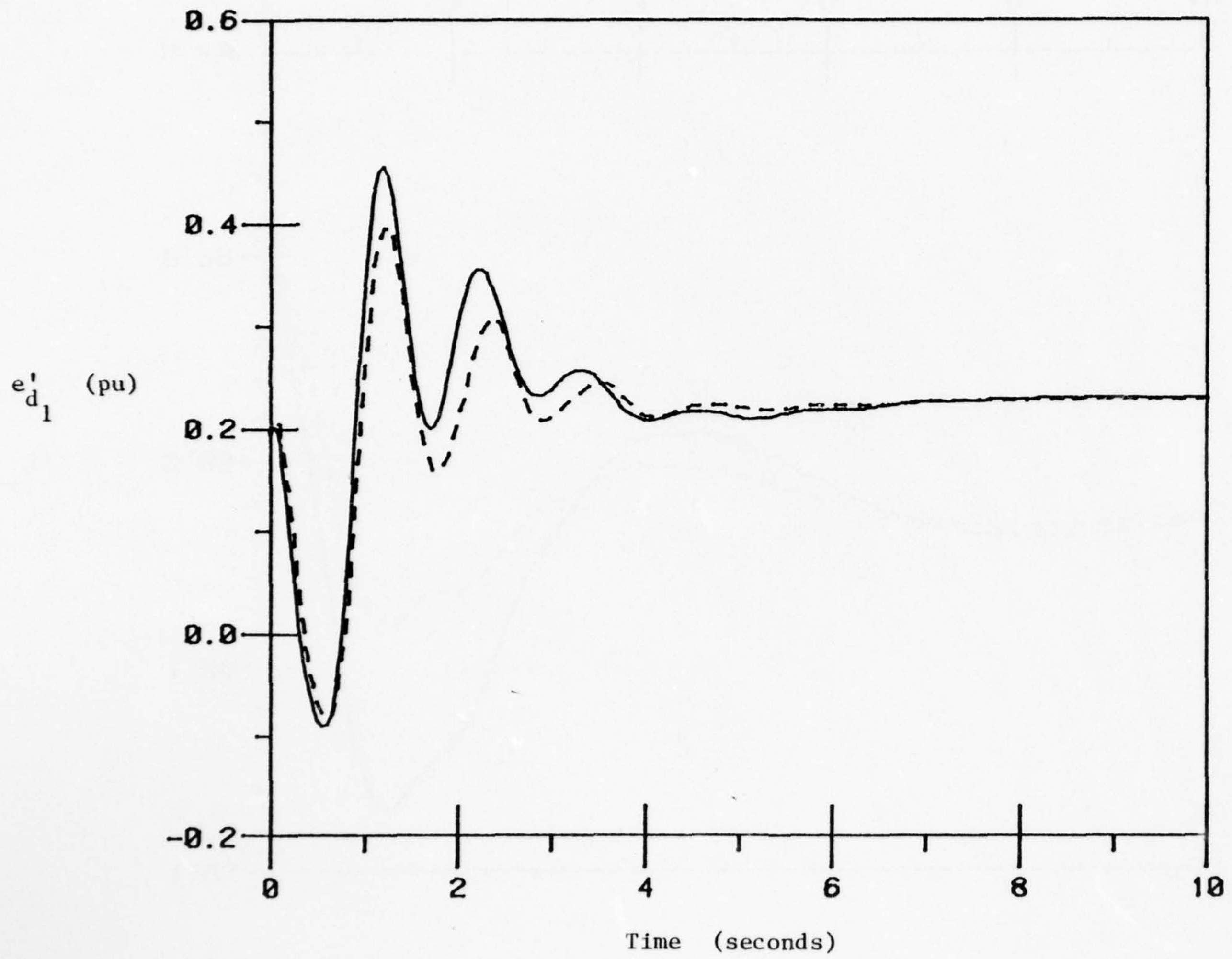


Fig. 4.50. Plots of exact and corrected approximation of e'_{d_1}.

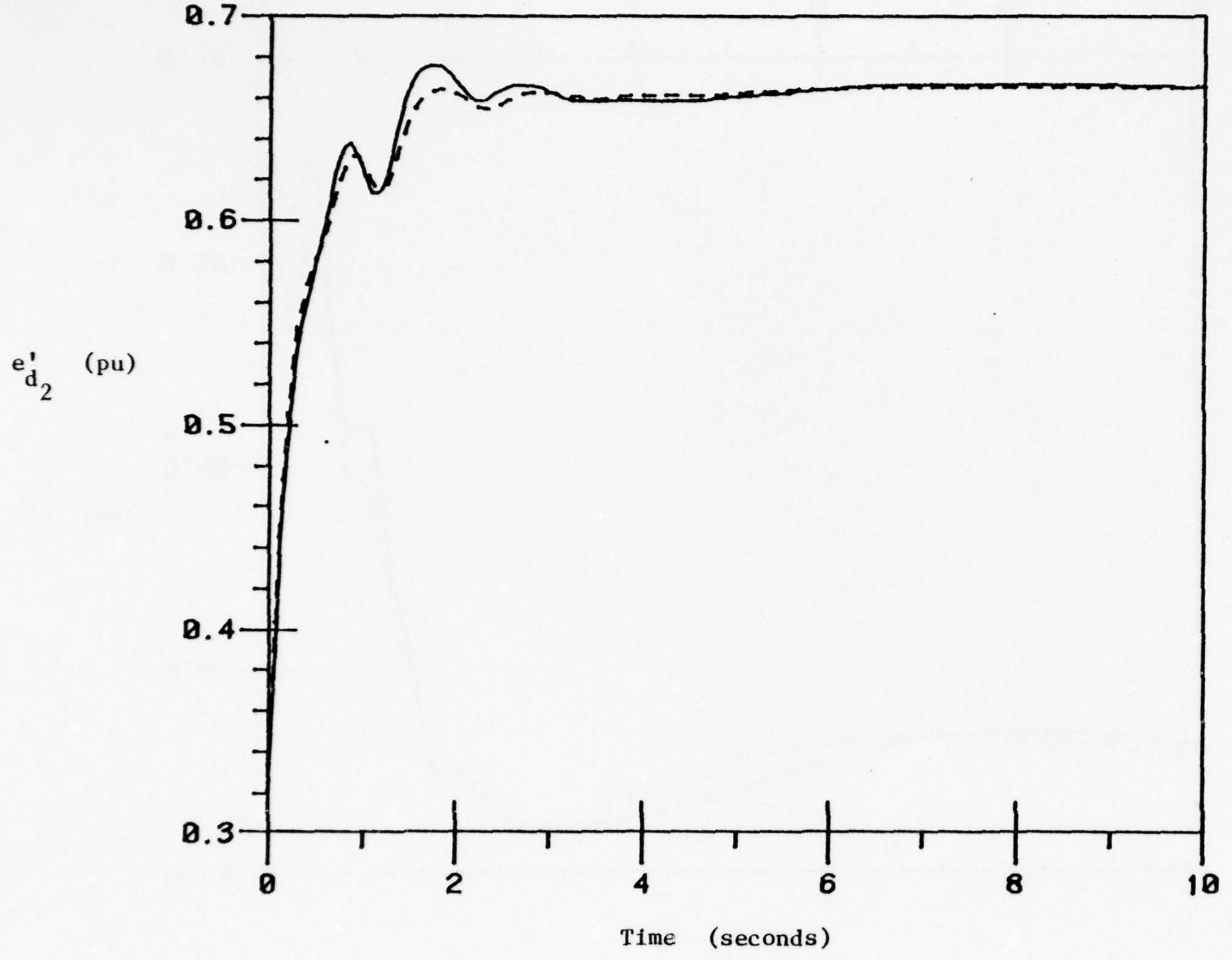


Fig. 4.51. Plots of exact and corrected approximation of e'_{d_2}.

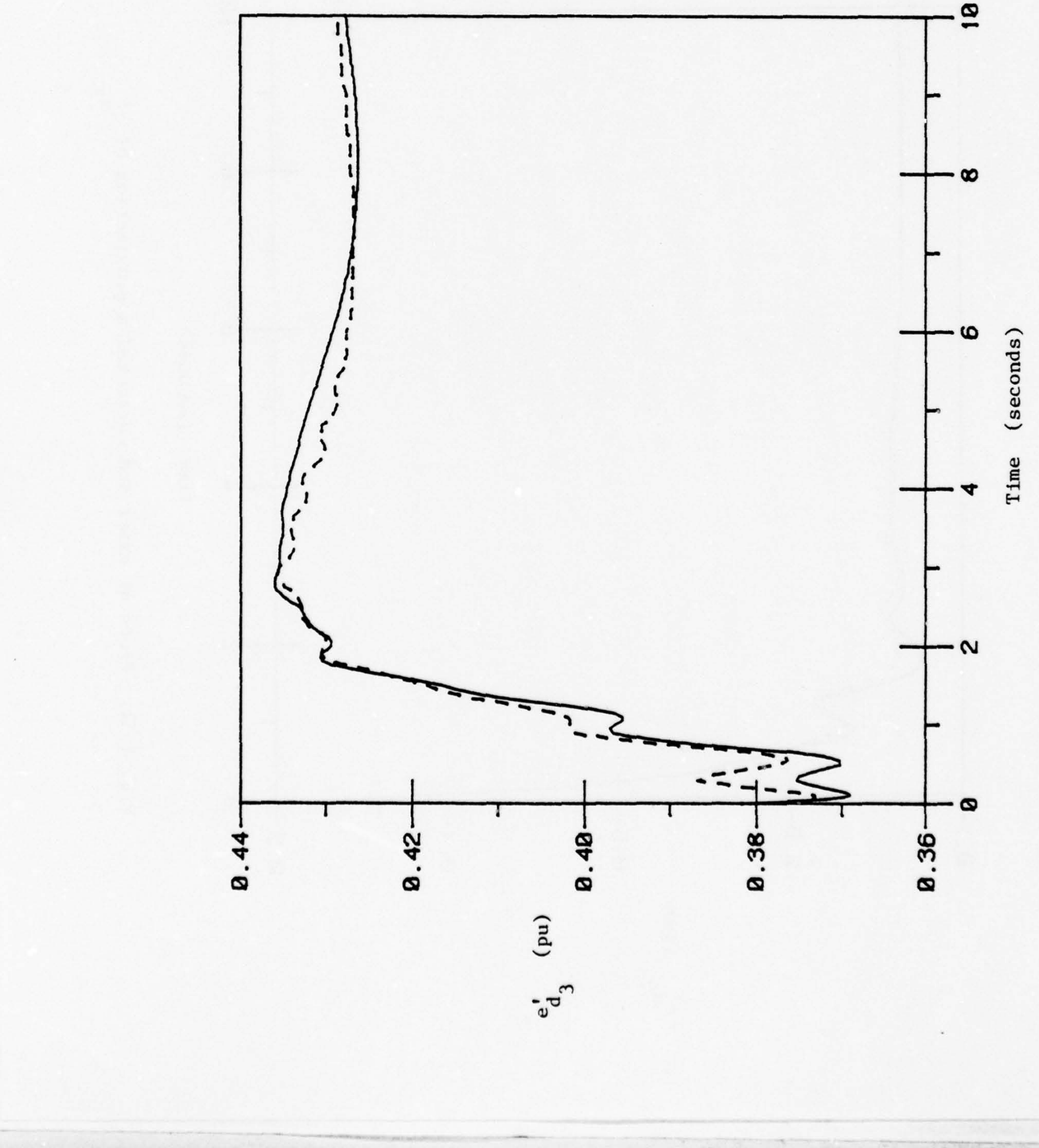


Fig. 4.52. Plots of exact and corrected approximation of e'_{d_3}.

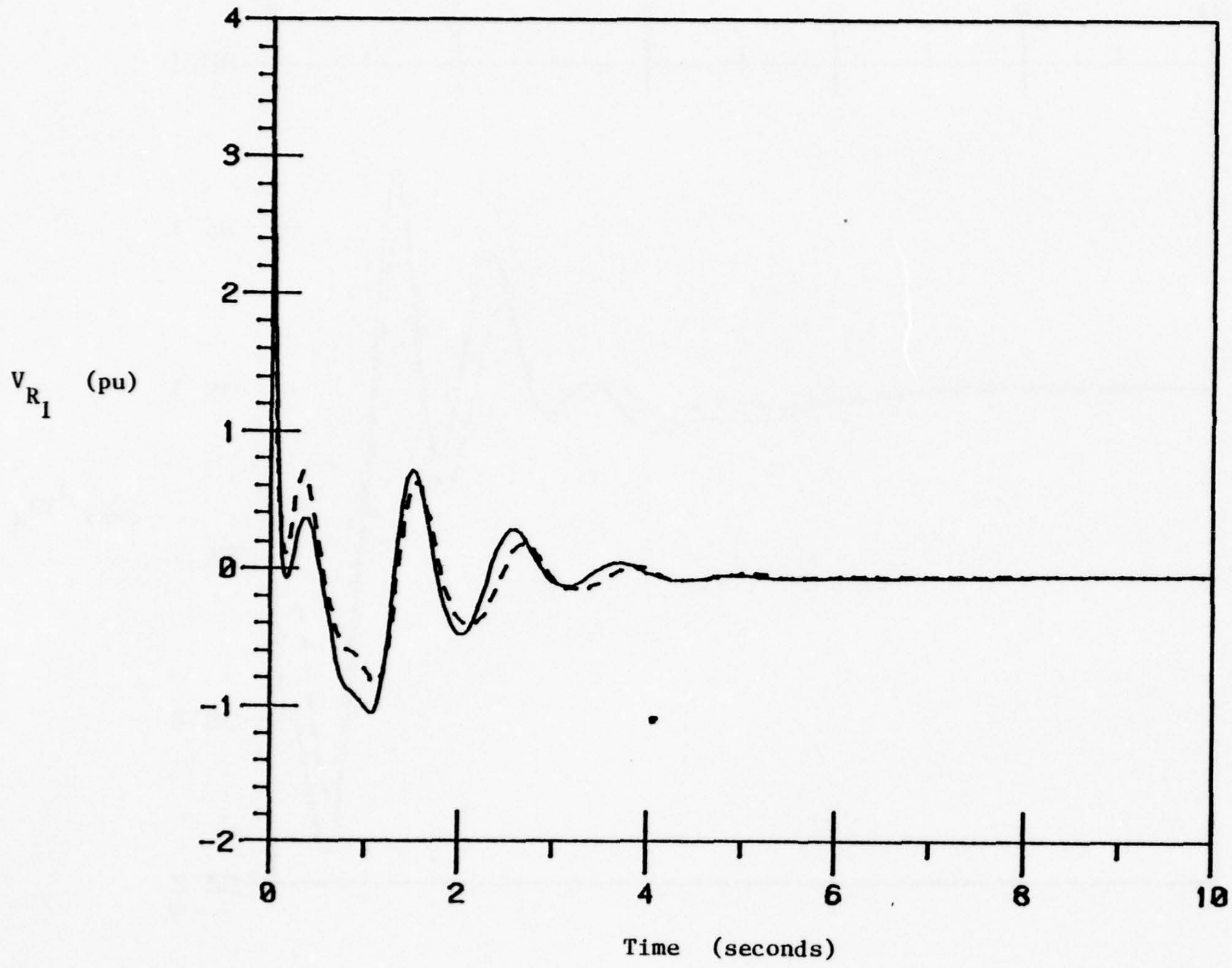


Fig. 4.53. Plots of exact and corrected approximation of V_{R_1}.

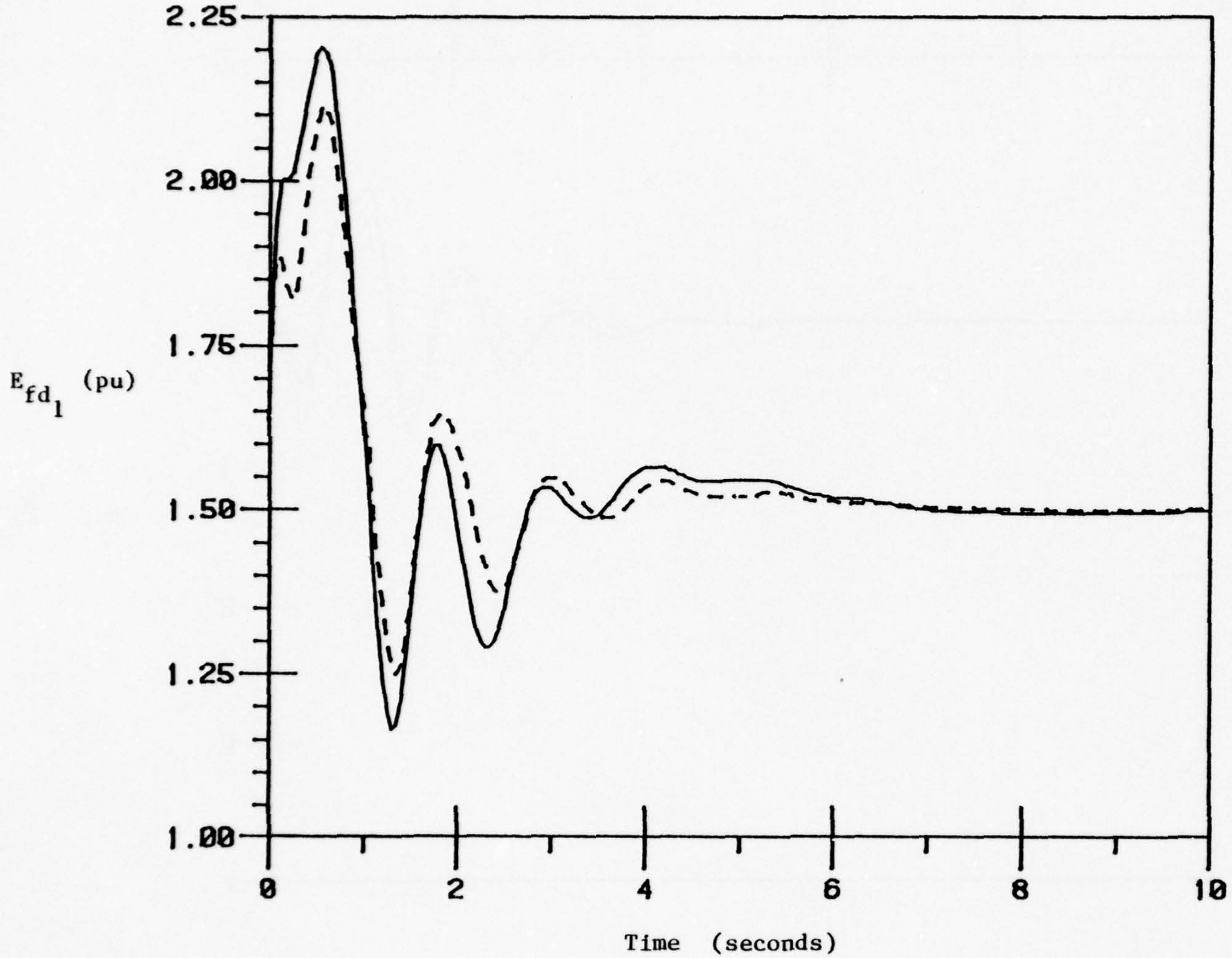


Fig. 4.54. Plots of exact and corrected approximation of E_{fd_1}.

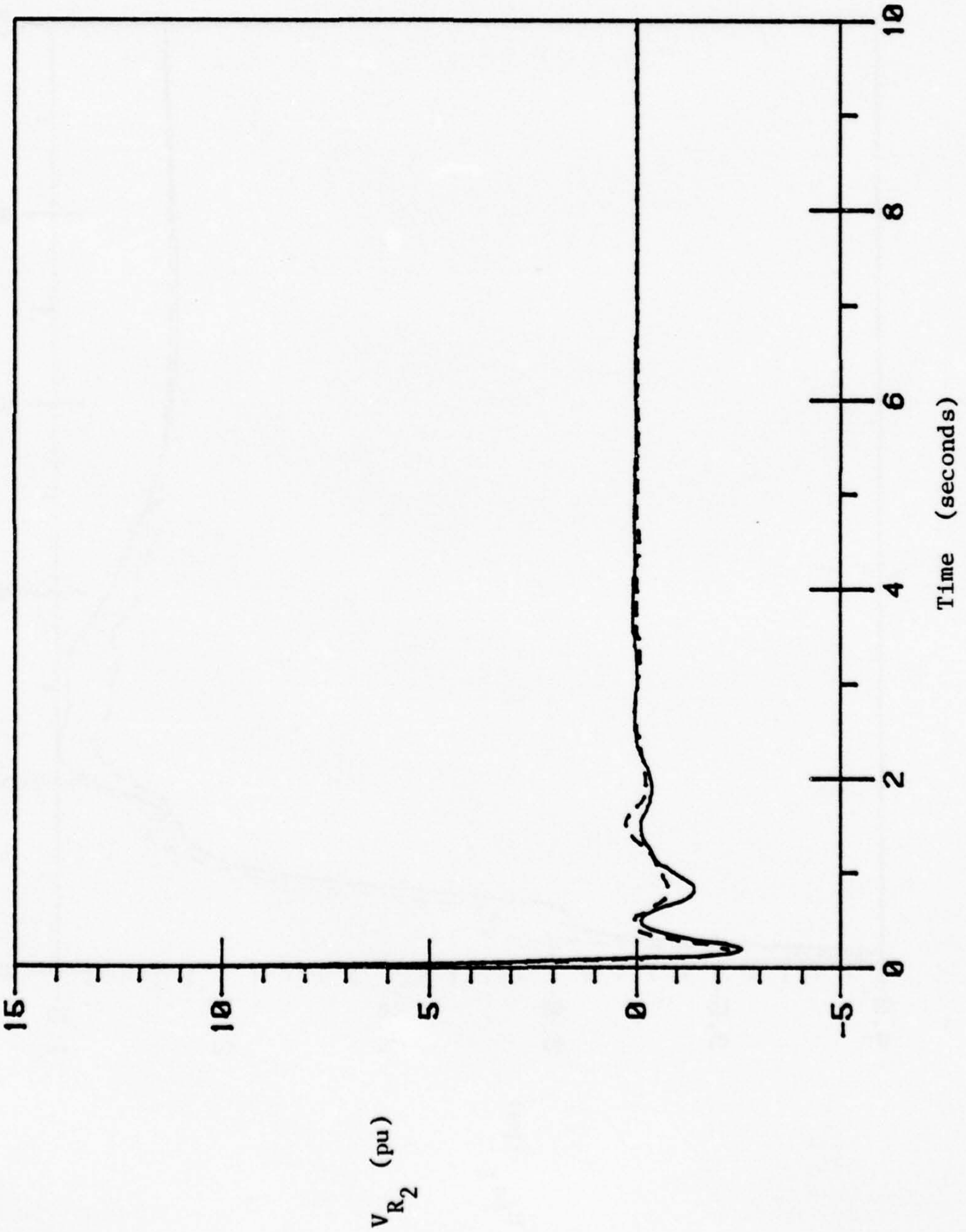


Fig. 4.55. Plots of exact and corrected approximation of V_{R_2}.

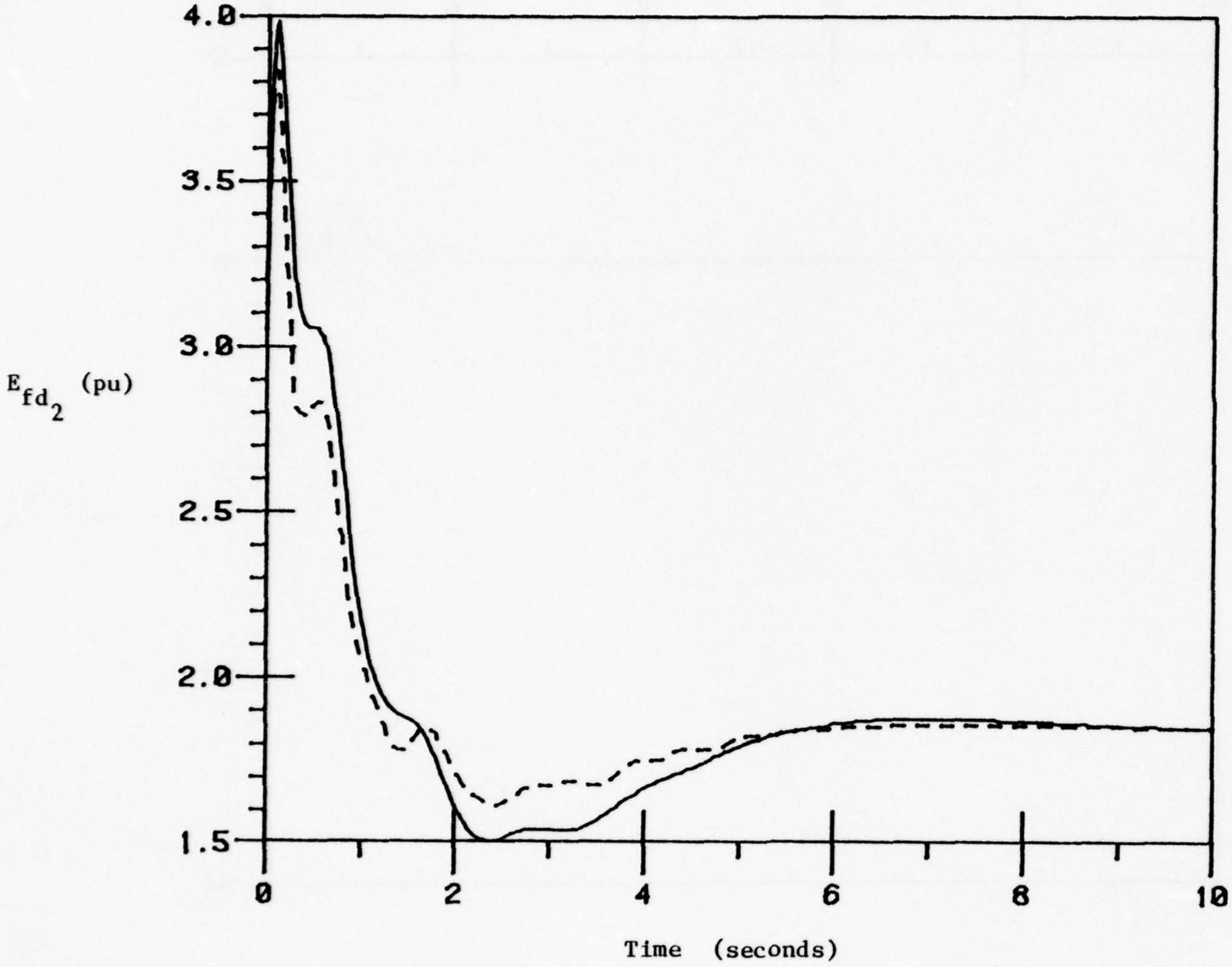


Fig. 4.56. Plots of exact and corrected approximation of E_{fd_2}.

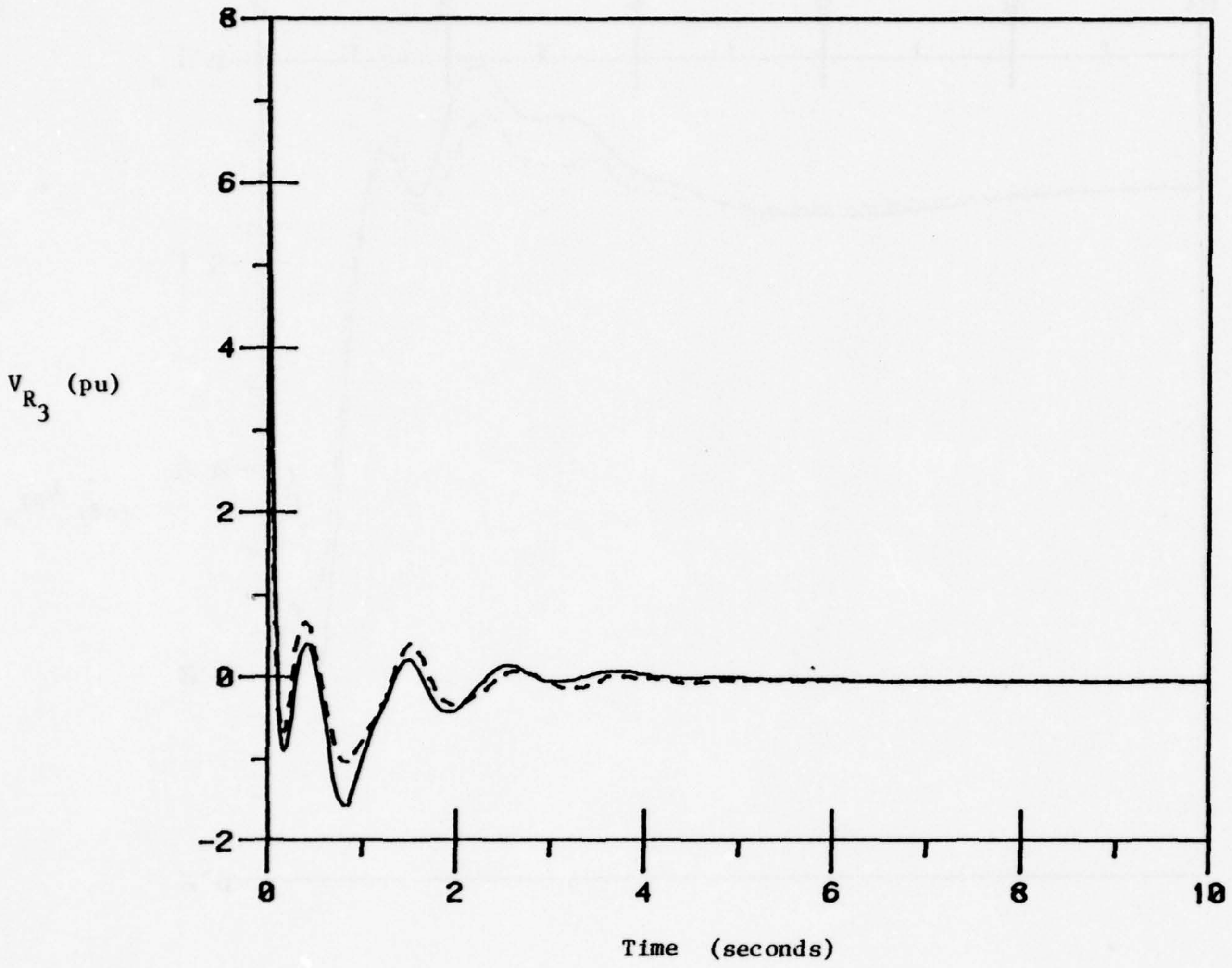


Fig. 4.57. Plots of exact and corrected approximation of V_{R_3}.

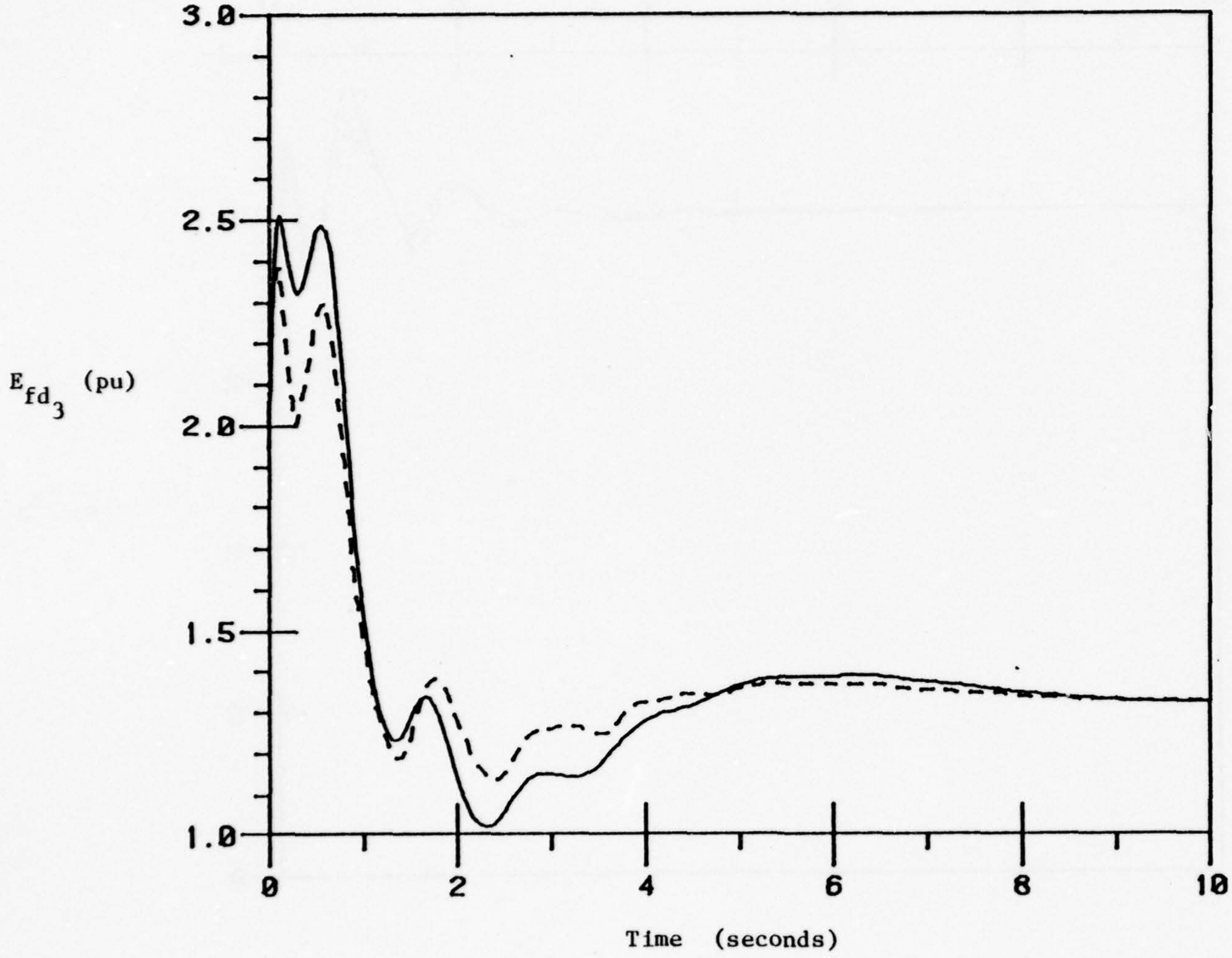


Fig. 4.58. Plots of exact and corrected approximation of E_{fd_3}.

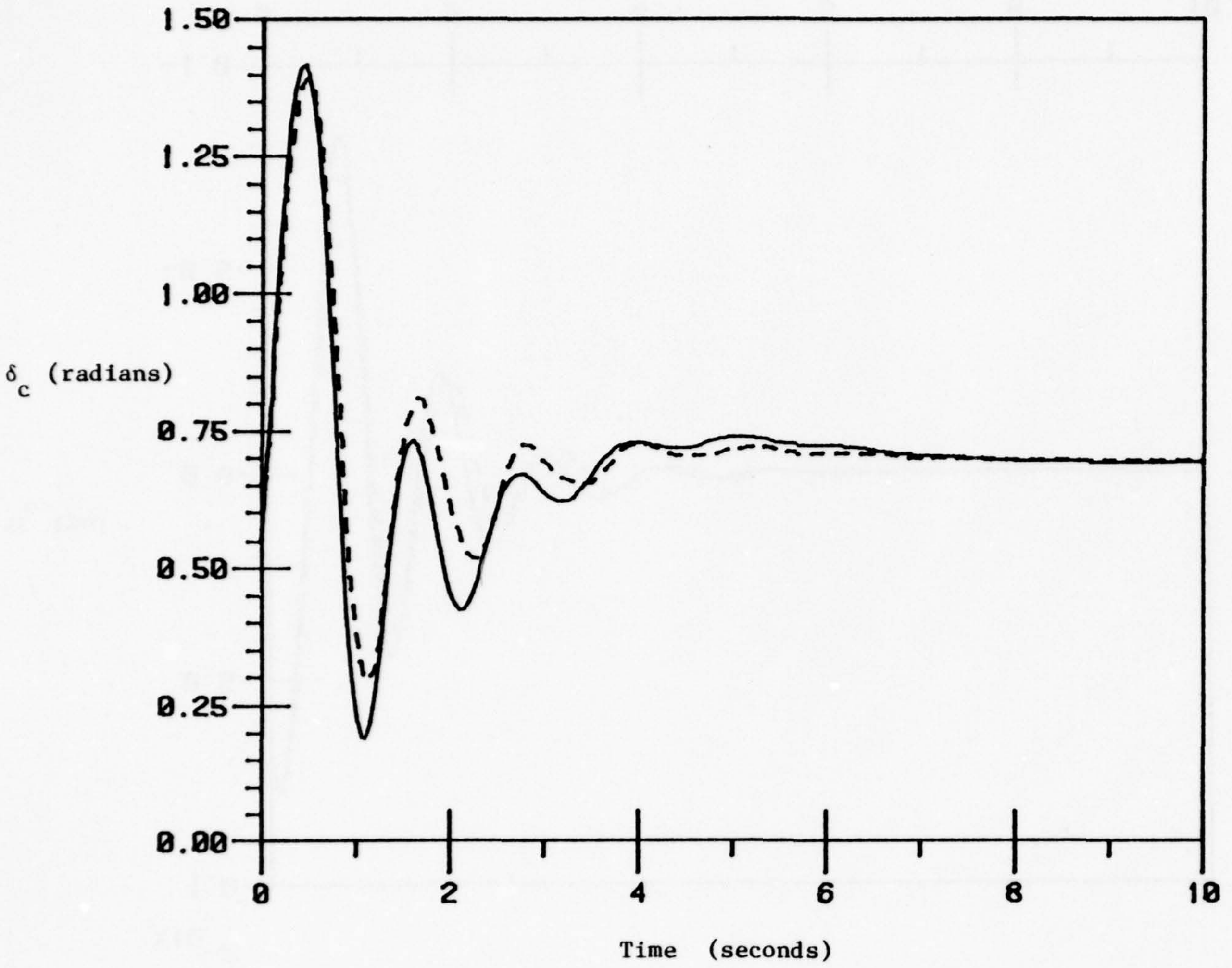


Fig. 4.59. Plots of exact and corrected approximation of δ_c.

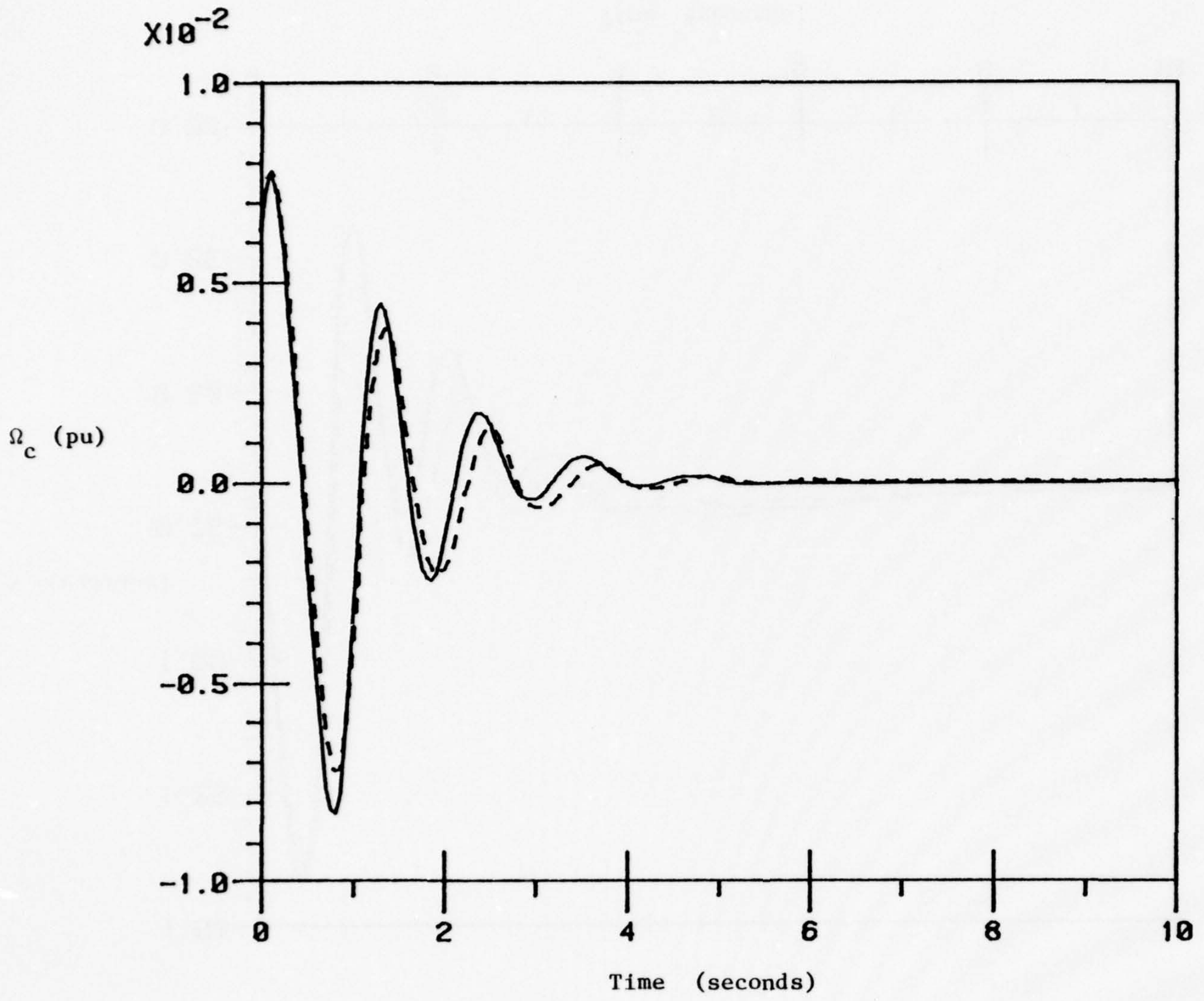


Fig. 4.60. Plots of exact and corrected approximation of Ω_c.

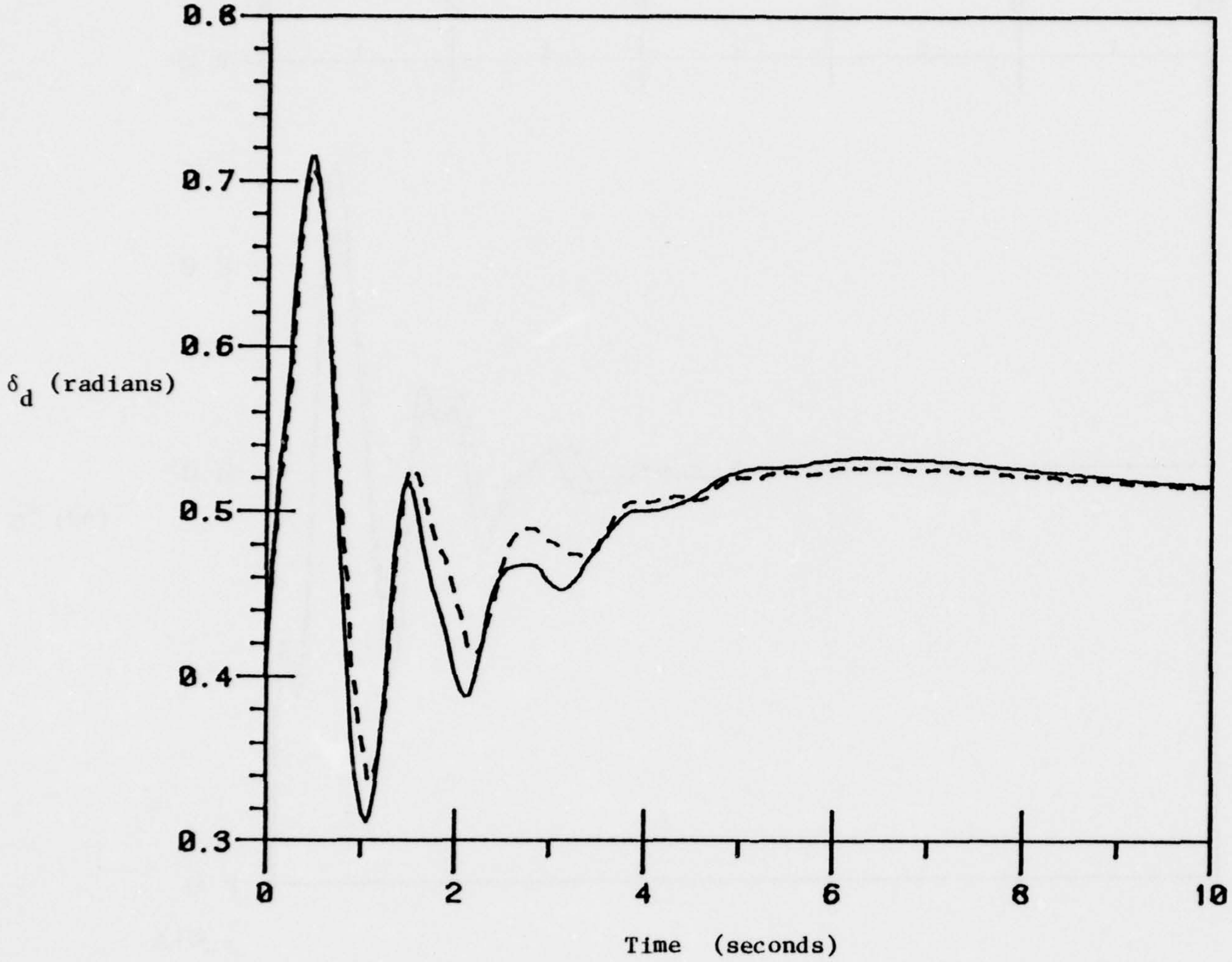


Fig. 4.61. Plots of exact and corrected approximation of δ_d.

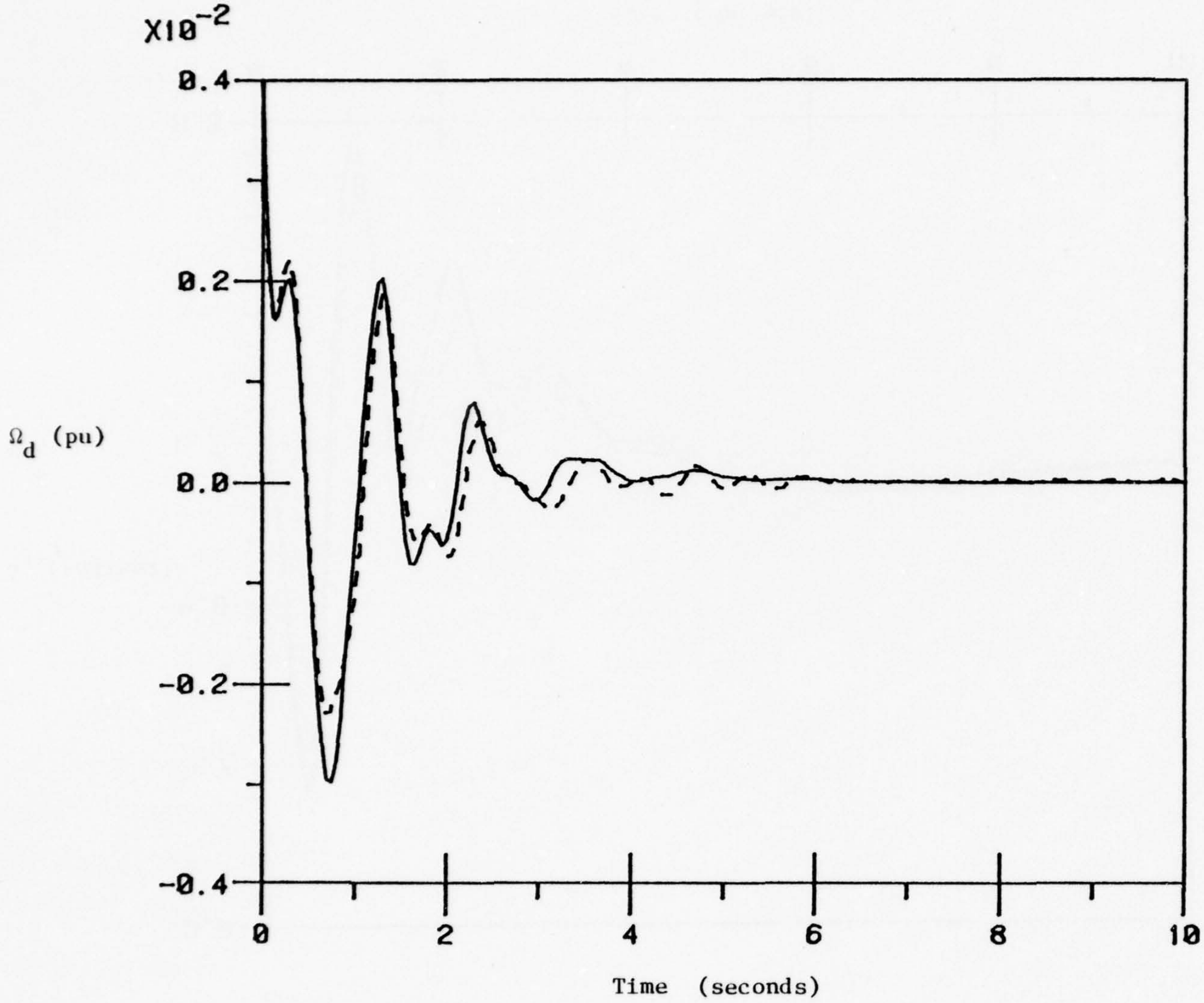


Fig. 4.62. Plots of exact and corrected approximation of Ω_d.

variables. In addition, it appears that the slow parts of some of the fast variables are more accurate due to the improvement of the slow subsystem approximation.

Thus, it is obvious that while this procedure has produced some improvement over the zero-order approximation, the improvement is not as dramatic as in the single machine case. The probable reason for this behavior is that the separation between the fast and slow dynamics is not as great in the multi-machine system as it is in the single machine system. We observed this fact when studying the linearized system. Hence, approximating the slow subsystem as an integrator of the fast dynamics is probably not very accurate. This state-of-affairs might be improved somewhat if e'_{d_3} were included with the slow subsystem.

4.6 Conclusions

In this chapter, we first explored eigenvalue location for two models of the three machine system, namely, 1) swing model and 2) full twentieth order model. Using the swing model, we were able to identify the system frequency drift mode and two rotor angle oscillatory modes. We found that the smaller of these modes was primarily determined by the combined inertia of two machines while the larger of the two was associated with the oscillation of one of these machines relative to the other. This observation prompted us to transform the angular variables to more readily display this behavior.

The eigenvalues of the full model were found to fall into groups the locations of which were very close to the locations of some of the eigenvalues of the single machine system. Hence we were able to identify which phenomena were responsible for each of the eigenvalue groups. We concluded that to achieve time scale separation, it is necessary to group the equations on a per phenomenon basis.

The variables Ω_r, e'_{q_1} ,e'_{q_2}, e'_{q_3}, R_{f_1}, R_{f_2}, and R_{f_3} were chosen to comprise the slow subsystem with all other variables in the fast subsystem. With this split, there is hardly any separation between the fastest of the slow modes and the slowest of the fast modes. Because of this lack of separation, the eigenvalue approximations took six iterations to converge to reasonable accuracy. The state approximations showed good accuracy after three iterations and excellent accuracy after four iterations.

As in the single machine case, the slow variables exhibited a large component of fast phenomena. Thus the zero-order non-linear approximation of the slow subsystem was not particularly accurate; however, the fast subsystem approximation was reasonably accurate. Applying three iterations of the non-linear correction procedure resulted in some improvement in the approximations. But this improvement was seen to be less dramatic than in the single machine probably because of the relatively small separation between the slow and fast subsystems.

5. CONCLUSION

This thesis applies singular perturbation techniques to a power system model describing rotor angle swings and associated phenomena.

First we considered a single machine-infinite bus system. Using the linearized model we ascertained the proper partitioning of the system into fast and slow subsystems. We also found that the eigenvalues and states of the linearized model could be approximated quite closely by using a block diagonalization procedure. Finally the zero order approximation of the non-linear model was investigated. The slow subsystem approximation was found to be in error because it failed to account for a large fast component in the slow variables due to forcing action of the field voltage. A method to correct for this inadequacy was introduced and was seen to perform well.

We concluded by studying a three machine system. From the linearized swing model, we found that this system possessed a slow mode which was not present in the single machine case, namely, the system frequency drift. In addition, the nature of the swing eigenvalues suggested a change of angle-speed variables in order to facilitate partitioning the system. Once this change of variables was made and after the equations were grouped on a per phenomenon basis, partitioning the system into fast and slow subsystems was relatively easy. Again we found the state and eigenvalue approximations of the linearized model to be very good, although more iterations were required in this case because of the relatively small separation between slow and fast subsystems. Finally the zero-order approximation of the twentieth order non-linear system was tested. The true slow subsystem responses were seen to have a large fast component as in the single machine case. We applied the correction procedure but with less dramatic success than in the single machine case probably because of the poor separation exhibited by this system.

Directions for future research are numerous. An automated methodology for partitioning an arbitrary system should be sought. More work needs to be done regarding correction methods for non-linear singularly perturbed systems. In this same light, models of some power system components such as voltage regulators and governors contain such "hard" non-linearities as limiters and deadbands. Methodologies need to be developed to account for these non-linearities.

APPENDIX A: Derivation of Synchronous Machine Model

Figure A1 shows a simplified three phase synchronous machine which illustrates some of the conventions and nomenclature which will be used in the following discussions.

Two circuits are shown on the rotor. The coil labelled f is the main field winding; the coil labelled g is a ficticious coil often introduced [16] to account for so-called "transient" effects in the quadrature axis. Its presence is an attempt to model eddy currents which can flow in the solid rotors of high speed turbo-alternators under non-steady state conditions. Additional rotor circuits are often included [9] to account for "subtransient" effects. These rotor circuits are needed in order to model eddy currents flowing near the surface of the rotor (or in the amortisseur winding of a salient pole machine) which contribute to the damping of rotor oscillations. Hence, the additional rotor circuits are often included when studying those problems in which an accurate assessment of rotor oscillation damping is considered essential. Since this thesis is concerned with time scale separation properties and not with accurate calculation of synchronous machine damping, these additional rotor circuits have not been included.

Notice also that the quadrature axis is shown in Fig. A1 to be leading the direct axis. This state-of-affairs was the formerly adopted convention [16]; but some recent literature [17] has suggested the opposite convention. Neither convention appears to have any particular advantage over the other.

Based on the discussion of the previous paragraphs, the following per unit equations for the machine's electrical variables can be written [18]

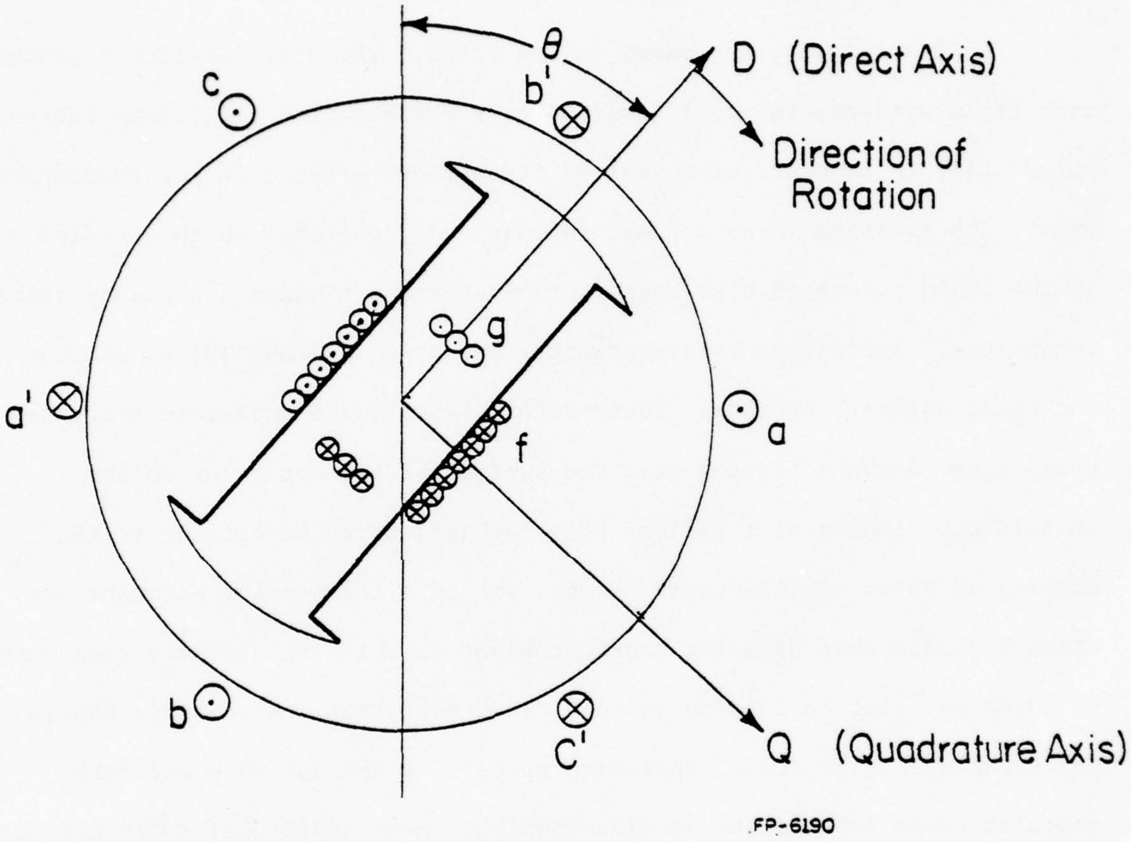


Fig. A.1. Simplified synchronous machine.

$$v_d = \frac{1}{\omega_0}\dot{\psi}_d - \Omega\psi_q \tag{A1a}$$

$$v_q = \frac{1}{\omega_0}\dot{\psi}_q + \Omega\psi_d \tag{A1b}$$

$$0 = r_g i_g + \frac{1}{\omega_0}\dot{\psi}_g \tag{A1c}$$

$$v_f = r_f i_f + \frac{1}{\omega_0}\dot{\psi}_f \tag{A1d}$$

$$\psi_d = -L_d i_d + M_{df}\, i_f \tag{A1e}$$

$$\psi_q = -L_q i_q + M_{qg}\, i_g \tag{A1f}$$

$$\psi_f = -M_{df} i_d + L_f\, i_f \tag{A1g}$$

$$\psi_g = -M_{qg}\, i_q + L_g\, i_g \quad . \tag{A1h}$$

The symbolism used here is explained in Appendix B.

Some additional comments concerning these equations are in order. First, the stator resistance has been neglected for simplicity. Second, the zero sequence equations are not included. The zero sequence relationships are only needed when unbalanced operation is being considered. All the studies carried out in this thesis are for balanced operation. Third, saturation of the synchronous machine is neglected. It has been found [18] that saturation can be approximately accounted for empirically. The usual procedure is to adjust the mutual inductances as a non-linear function of the currents. One of these approximate methods could have been employed in this thesis; however, because our main emphasis is studying time scale separation, we chose the simplest synchronous machine model which nevertheless retains the basic time scale information.

Now we carry out some manipulations and simplifications to put the general equations (A1) in their final form. First the $\dot{\psi}_d$ and $\dot{\psi}_q$

terms in (Ala) and (Alb) are neglected and Ω is set to Ω_0 in the same equations. The former step is almost always taken [19]. The inclusion of these terms accounts for certain rapidly decaying components of the stator current during transients. The torque produced by these components of the current is generally negligible except in the case of a fault on the generator terminals in which case the neglected components of the braking torque can be accounted for empirically. A more compelling reason to neglect these terms is that if these terms were included in the stator equations, then similar terms arising from inductances in the network to which the machine is connected should be included. For large networks, this requirement is clearly prohibitive.

The latter simplification carried out above is often done [19]. As pointed out in [20] both the terminal voltage and network inductive reactance vary in direct proportion to the machines' frequencies. Hence, in a network which is mainly inductive, relatively small variations in machine speed do not appreciably effect the calculation of network currents.

The modified stator voltage equations are

$$v_d = -\Omega_0\psi_q \tag{A2a}$$

$$v_q = \Omega_0\psi_d \ . \tag{A2b}$$

We substitute (Alf) into (A2a) and (Ale) into (A2b) with the result

$$v_d = \Omega_0 L_q i_q - \Omega_0 M_{qg} i_g \tag{A3a}$$

$$v_q = -\Omega_0 L_d i_d + \Omega_0 M_{df} i_f \ . \tag{A3b}$$

Now solve (Alh) for i_g and (Alg) for i_f yielding

$$i_g = \frac{1}{L_g}\psi_g + \frac{M_{qg}}{L_g} i_q \tag{A4a}$$

$$i_f = \frac{1}{L_f}\psi_f + \frac{M_{df}}{L_f} i_d \quad . \tag{A4b}$$

Finally substitute (A4a) into (A3a) and (A4b) into (A3b) to give

$$v_d = \Omega_0 L'_q i_q - \Omega_0 \frac{M_{qg}}{L_g}\psi_g \tag{A5a}$$

$$v_q = -\Omega_0 L'_d i_d + \Omega_0 \frac{M_{df}}{L_f}\psi_f \tag{A5b}$$

where

$$L'_d = L_d - \frac{M^2_{df}}{L_f} \tag{A6}$$

is the direct axis transient inductance and

$$L'_q = L_q - \frac{M^2_{qg}}{L_g} \tag{A7}$$

is the quadrature axis transient inductance.

At this point we let

$$x'_d = \Omega_0 L'_d \tag{A8a}$$

$$x'_q = \Omega_0 L'_q \tag{A8b}$$

$$e'_d = -\Omega_0 \frac{M_{qg}}{L_g}\psi_g \tag{A8c}$$

$$e'_q = \Omega_0 \frac{M_{df}}{L_f}\psi_f \quad . \tag{A8d}$$

The first two quantities above are respectively the direct axis and quadrature axis reactances. Thus using (A8), (A5) can be written in final form as

$$v_d = x'_q i_q + e'_d \tag{A9a}$$

$$v_q = -x'_d i_d + e'_q \quad . \tag{A9b}$$

We will discuss the role of (A9) after we derive the remaining equations for this synchronous machine model.

First we obtain the differential equations for e'_d and e'_q. Substituting (A4a) into (Alc) gives

$$\frac{1}{\omega}\frac{d\psi_g}{dt} + \frac{r_g}{L_g}\psi_g + \frac{r_g}{L_g}M_{qg}i_q = 0 \ . \tag{A10}$$

Multiply this equation by $-\Omega_0 M_{qg}/L_g$. Then by using the definitions of e'_d and L'_q, the result can be written as

$$\frac{1}{\omega_0}\frac{de'_d}{dt} + \frac{r_g}{L_g}e'_d - \frac{r_g}{L_g}\Omega_0(L_q-L'_q)i_q = 0 \quad . \tag{A11}$$

The ratio r_g/L_g is the per unit quadrature axis open circuit time constant, $T'_{q_{0(pu)}}$. We can also make the substitution

$$x_q - x'_q = \Omega_0(L_q-L'_q) \quad . \tag{A12}$$

With these substitutions, (All) becomes

$$\frac{1}{\omega_0}\frac{de'_d}{dt} = \frac{1}{T'_{q_{0(pu)}}}\left[- e'_d + (x_q-x'_q)i_q\right] \quad . \tag{A13}$$

Finally, we multiply both sides of (A13) by ω_0. The ratio $\omega_0/T'_{q_{0(pu)}}$ is $1/T'_{q_0}$ if T'_{q_0} is expressed in seconds. Thus, the final form of the differential equation for e'_d is

$$\frac{de'_d}{dt} = \frac{1}{T'_{q_0}}\left[- e'_d + (x_q-x'_q)i_q\right] \quad . \tag{A14}$$

Carrying out similar manipulations on (A1d) gives the following differential equation for e'_q

$$\frac{de'_q}{dt} = \frac{1}{T'_{d_0}}[-e'_q - (x_d - x'_d)i_d] + \omega_0\Omega_0 \frac{M_{df}}{L_f} v_f \ . \tag{A15}$$

This equation (or one similar to it) is often written in a slightly different form which we now obtain.

In the per unit system being used in this thesis [21], base field current is chosen so that one per unit field current establishes the same fundamental component of the mutual flux as does one per unit stator current. In other words, the base field current is chosen such that

$$\frac{M_{df}}{v_{a(b)}/\omega_0 i_{f(b)}} = \tilde{M}_{df} \tag{A16}$$

where the tilde is temporarily used to denote per unit quantities and the subscript "b" indicates base quantities. $\tilde{M}_{df}$ is the difference between the per unit direct axis inductance and the per unit stator leakage inductance. It is common practice [13] to choose the base exciter voltage $E_{fd(b)}$ so that one per unit exciter voltage produces one per unit open circuit terminal voltage on the air gap line of the synchronous machine. From (A3b) the open circuit voltage is

$$v_q = \omega_0 M_{df} i_f \ . \tag{A17}$$

Note that (A17) is in ordinary units (volts, amps, etc). Hence, we set

$$v_{a(b)} = \omega_0 M_{df} \frac{E_{fd(b)}}{r_f} \tag{A18}$$

in other words

$$\frac{v_{a(b)}}{E_{fd(b)}} = \frac{\omega_0 M_{df}}{r_f} \quad . \tag{A19}$$

But from (A16) we have

$$\omega_0 M_{df} = \frac{v_{a(b)}}{i_{f(b)}} \tilde{M}_{df} \tag{A20}$$

and from the definition of per unit quantities

$$r_f = \tilde{r}_f \frac{v_{f(b)}}{i_{f(b)}} \quad . \tag{A21}$$

Substituting (A20) and (A21) into (A19) gives upon rearrangement

$$\frac{v_{f(b)}}{E_{fd(b)}} = \frac{\tilde{M}_{df}}{\tilde{r}_f} \quad . \tag{A22}$$

Now let the voltage applied to the field circuit be E_{fd}(volts). This value can be written as the product of its per unit value and the base exciter voltage

$$E_{fd} = \tilde{E}_{fd} E_{fd(b)} \cdot \tag{A23}$$

To put this quantity in per unit on the base field voltage, we divide (A23) by $v_{f(b)}$ to give

$$\frac{E_{fd}}{v_{f(b)}} = \tilde{E}_{fd} \frac{E_{fd(b)}}{v_{f(b)}} \quad . \tag{A24}$$

Substituting (A22) into (A24) and using $\tilde{T}'_{d_0} = \tilde{L}_f/\tilde{r}_f$ yield

$$\frac{E_{fd}}{v_{f(b)}} = \tilde{E}_{fd} \frac{\tilde{L}_f}{\tilde{M}_{df}} \frac{1}{\tilde{T}'_{d_0}} \quad . \tag{A25}$$

In other words, a per unit value of E_{fd} (which might come from the solution of the exciter equations) must be multiplied by $\tilde{L}_f/\tilde{M}_{df}\tilde{T}'_{d_0}$ in order to refer it to the field circuit.

We now return to (A15) and hereafter drop the tilde convention for per unit quantities. We substitute the per unit value of E_{fd} from (A25) into (A15) to give

$$\frac{de'_q}{dt} = \frac{1}{T'_{d_0}}[-e'_q - (x_d - x'_d)i_d + E_{fd}] \tag{A26}$$

where all quantities are in per unit except T'_{d_0} which is in seconds. This final form follows since $\Omega_0 = 1$ per unit.

We next consider the mechanical equation of motion which can be written as [18]

$$2H\frac{d\Omega}{dt} = T_{in} - T_{ele} - D(\Omega-1) \ . \tag{A27}$$

The last term on the right hand side of (A27) is a damping term which is often introduced [9] to account for such effects as changes in turbine power with speed and changes in connected load with system frequency. The developed electrical torque is [18]

$$T_{ele} = \psi_d i_q - \psi_q i_d \quad . \tag{A28}$$

Now if (A4b) is combined with (A1e) ψ_d can be written as

$$\psi_d = -L'_d i_d + \frac{M_{df}}{L_f}\psi_f \quad . \tag{A29}$$

Similarly, ψ_q is

$$\psi_q = -L'_q i_q + \frac{M_{qg}}{L_g}\psi_g \quad . \tag{A30}$$

Using (A29) and (A30) together with (A8c) and (A8d) in (A28) gives the following result for the electrical torque

$$T_{ele} = (L_q' - L_d')i_d i_q + \frac{1}{\Omega_0}(e_q' i_q + e_d' i_d)$$

$$= (L_q' - L_d')i_d i_q + e_q' i_q + e_d' i_d \qquad \text{(A31)}$$

since $\Omega_0 = 1$ per unit. The final form for (A27) is obtained by substituting

$$T_{in} = \frac{P_{in}}{\Omega} \ , \qquad \text{(A32)}$$

hence,

$$2H \frac{d\Omega}{dt} = \frac{P_{in}}{\Omega} - (L_q' - L_d')i_d i_q - e_q' i_q - e_d' i_d - D(\Omega - 1) \quad . \qquad \text{(A33)}$$

Having derived the differential equations for the synchronous machine model, the importance of the currents, i_d and i_q, is clearly seen. Equation (A9) is one relationship between the machine terminal voltage and current. There is another constraint between the voltage and current, namely the constraint imposed by the network to which the machine is connected. Therefore, the procedure for computing the currents in terms of the state variables depends on how the network is modelled.

A condition which results from neglecting the $\dot{\psi}_d$ and $\dot{\psi}_q$ terms in (A1a) and (A1b) is that the network currents and voltages are in a quasi-sinusoidal steady state. The reason we say "quasi-steady state" is that the per unit frequency of the voltages produced by the synchronous machines is equal to their per unit speed. Since these speeds vary during a transient, the currents and voltages cannot truly be considered as sinusoidal. This non-sinusoidal character is always neglected in large system studies.

If the currents and voltage are considered to be sinusoidal, then they can be represented as phasors. The frequency of these sinusoids is taken to be the fundamental frequency (60 Hz). In general, the only speed effect which might be included is the variation of the magnitude of the machine's terminal voltage with speed.

Regarding the phasors, the formerly accepted practice [16] was to take direct axis quantities as real quantities and quadrature axis quantities as imaginary quantities. More recently [17], the opposite convention has been advocated. In this thesis, we will use the former convention. For example, the phasor for the machine's terminal voltage is written $v_d + jv_q$. Note that phasors written on this basis are local to a particular machine because they depend on the position of the machine's direct axis for reference. The implication of this observation is that the angular differences between the machines' direct axes are fundamental in calculating the network currents. The variation of these angular differences during transients is therefore of great importance in assessing system stability following disturbances.

A vast simplification in computing the machine's currents is obtained if transient saliency is neglected; that is, we let $x'_d = x'_q = x'$. Hereafter, we assume that this has been done. Note that (A9) can now be written as one equation

$$e'_d + je'_q = jx'(i_d + ji_q) + v_d + jv_q \quad . \tag{A34}$$

Equation (A34) is sometimes referred to as the voltage behind transient reactance model. Using (A34) allows us to easily compute the components of the currents in terms of the state variables because the transient reactance can be combined with any external impedance.

As mentioned previously, the task of calculating the currents depends very much on how the network is modelled. In this thesis, we represent the loads by constant impedance. With this simplification, the network can be reduced so that only the internal generator nodes remain. Then the generator currents can be written as

$$\overline{I}_{(1)} = \overline{Y}\,\overline{V}_{(1)} \tag{A35}$$

where $\overline{I}_{(1)}$ and $\overline{V}_{(1)}$ are vectors the components of which are the complex generator currents and voltages respectively. The subscript 1 indicates that the reference for these phasors is the direct axis of machine 1. It is easy to show that any machine phasor, say $\overline{\alpha}$, which uses the machine 1 direct axis as reference can be ref:rred to its own direct axis by applying the transformation

$$\overline{\alpha}_{(m)} = e^{-j\delta_{j1}}\,\overline{\alpha} \tag{A36}$$

where δ_{j1} is the angle between the direct axes of machine j and machine 1 and the subscript m denotes machine reference.

Now, write

$$\overline{I}_{(1)} = \overline{T}\,\overline{I}_{(m)} \tag{A37a}$$

$$\overline{V}_{(1)} = \overline{T}\,\overline{V}_{(m)} \tag{A37b}$$

where

$$\overline{T} = \begin{bmatrix} 1+j0 & 0 & 0 \;.\;.\;. & 0 \\ 0 & e^{j\delta_{21}} & 0 \;.\;.\;. & 0 \\ \vdots & & & \\ 0 & 0 & .\;\;.\;\;. & e^{j\delta_{n1}} \end{bmatrix} . \tag{A38}$$

Substituting (A37) into (A35) gives

$$\begin{aligned}\bar{I}_{(m)} &= \bar{T}^{-1}\bar{Y}\bar{T}\bar{V}_{(m)} \\ &= \bar{T}^{*}\bar{Y}\bar{T}\bar{V}_{(m)} \\ &= \hat{\bar{Y}}\,\bar{V}_{(m)} \end{aligned} \tag{A39}$$

where the superscript * indicates complex conjugate. If we write

$$\bar{Y}_{ij} = Y_{ij}\, e^{j\theta_{ij}} \tag{A40}$$

then the elements of $\hat{\bar{Y}}$ are

$$\hat{\bar{Y}}_{ij} = Y_{ij}\, e^{j(\theta_{ij} + \delta_{j1} - \delta_{i1})} \quad . \tag{A41}$$

Splitting (A39) into its real and imaginary parts gives

$$i_d + ji_q = (\hat{Y}_r + j\hat{Y}_i)(e'_d + je'_q) \quad . \tag{A42}$$

In (A42), i_d, i_q, e'_d, and e'_q are vectors; the subscript r denotes "real part of" and the subscript i denotes "imaginary part of." Equating real and imaginary parts on both sides of (A42) yields

$$i_d = \hat{Y}_r e'_d - \hat{Y}_i e'_q \tag{A43a}$$

$$i_q = \hat{Y}_r e'_q + \hat{Y}_i e'_d \quad . \tag{A43b}$$

The i-th component of (A43a) can be written as

$$i_{d_i} = \sum_{j=1}^{n} (\hat{Y}_{r_{ij}} e'_{d_j} - \hat{Y}_{i_{ij}} e'_{q_j}) \tag{A44}$$

which, if we use (A41), becomes

$$i_{d_i} = \sum_{j=1}^{n} Y_{ij}[e'_{d_j}\cos(\theta_{ij} + \delta_{j1} - \delta_{i1}) - e'_{q_j}\sin(\theta_{ij} + \delta_{j1} - \delta_{i1})]. \tag{A45}$$

where n is the number of generator nodes. Similarly for (A43b) we get

$$i_{q_i} = \sum_{j=1}^{n} Y_{ij}[e'_{q_j}\cos(\theta_{ij}+\delta_{j1}-\delta_{i1})+e'_{d_j}\sin(\theta_{ij}+\delta_{j1}-\delta_{i1})]. \qquad \text{(A46)}$$

From (A45) and (A46) the importance of the angular differences is clearly seen. These angles must be included in the state description of any system of synchronous machines. Differentiating the angle δ_{j1} with respect to time gives

$$\begin{aligned}\dot{\delta}_{j1} &= \omega_0(\Omega_j - \Omega_1)\\ &= 377\ (\Omega_j - \Omega_1), \quad j = 2,3,\cdots,n\ . \qquad \text{(A47)}\end{aligned}$$

The factor 377 is needed to convert from per unit to radians. With (A47) we complete the derivation of the general equations for the synchronous machine model.

The single machine-infinite bus system shown in Fig. 3.1 allows some simplification of (A45) and (A46). Choosing $\overline{V}_i$ as the reference and writing

$$\overline{Y} = \frac{1}{j(x'+x_e)} \quad , \qquad \text{(A48)}$$

it is easy to show that

$$i_d = Y(e'_q + V_i \sin\delta) \qquad \text{(A49a)}$$

$$i_q = Y(-e'_d + V_i \cos\delta)\ . \qquad \text{(A49b)}$$

The other equations needed for both the single machine and multi-machine models are the equations describing the voltage regulator. By inspection of Fig. 3.2, these equations are

$$\dot{R}_f = \frac{1}{T_F}\,(-R_f + \frac{K_F}{T_F}\,E_{fd}) \qquad \text{(A50a)}$$

$$\dot{E}_{fd} = \frac{1}{T_E}\,\{-[K_E + S_E(E_{fd})]E_{fd} + V_R\} \qquad \text{(A50b)}$$

$$\dot{V}_R = \frac{1}{T_A}\,[K_A(R_f - \frac{K_F}{T_F}\,E_{fd} - V_t + V_{Ref}) - V_R] \quad . \qquad \text{(A50c)}$$

In (A50c), V_t is the magnitude of the machine's terminal voltage. Since the terminal voltage is $v_d + jv_q$, the magnitude is

$$V_t = \sqrt{v_d^2 + v_q^2} \quad . \qquad \text{(A51)}$$

APPENDIX B: Notation

With reference to the synchronous machine electrical equations of Appendix A, the following notation has been adopted:

Subscripts

d - direct axis quantities

q - quadrature axis quantities

f - main field quantities

g - quadrature axis "field" quantities

Other Symbols

i - current

L - self inductance

M - mutual inductance

r - resistance

v - voltage

ψ - flux linkage

ω_0 - machine rated electrical speed in radians/second

Ω - machine per unit electrical speed

APPENDIX C: Listing of Computer Programs

This appendix gives the listings of the three major programs used during this research.

1. Program LINSP does the calculations for a linear singularly perturbed system.
2. Program VRSMSP does the calculations for the non-linear model of the single machine-infinite bus system.
3. Program MMSP does the calculations for the non-linear model of the multi-machine system.

These programs call the following system library subroutines:

1. RKGS - from SSP for integrating differential equations.
2. LEQT1F - from IMSL for solving linear equations.
3. EIGRF - from IMSL for calculating eigenvalues.
4. INITT
5. BINITT
6. NPTS
7. XFRM
8. YFRM
9. CHECK
10. DSPLAY
11. FRAME
12. ANMODE
13. CPLOT
14. LINE
15. DLIMY

From Tektronix Software Package for drawing graphs.

```
        !********PROGRAM LINSP PAGE 1********
        DIMENSION FA(20,20),A(7,7),B(7,13),CO(13,7),D(13,13),
        1W(20),CN(13,7),WIC(20),PRMT(5),AUX(8,20),AL(13,7),
        2H(7,13),WDOT(20),DS(13,13),WKAREA(13),AS(7,7),BN(7,
        313),TR(20,20),TRI(20,20),XI(7),ETA(13),PLT(501),TFST
        4(501),TSLO(501),XIS(7,501),ETAS(13,501),WH(501),WS(20,
        5501),TW(501),DI(13,13)
        COMPLEX UA(7),UD(13),Z(1,1)
        COMMON/BLK1/FA,N
        COMMON/BLK2/A,NS
        COMMON/BLK3/D,NF
        COMMON/BLK4/WS,TW,NSW
        COMMON/BLK5/XIS,TSLO,NSS
        COMMON/BLK6/ETAS,TFST,NSF
        EXTERNAL FW,OW,FSLO,OSLO,FFST,OFST
1       FORMAT(A1)
4       FORMAT(A4)
11      FORMAT(I)
31      FORMAT(G)
40      FORMAT(' ??',A1,'?? TYPE AGAIN ',$)
100     TYPE 200
200     FORMAT(' GIVE A MATRIX FILE NAME ',$)
        ACCEPT 4,NNAM
        TYPE 300
300     FORMAT(' NS= ',$)
        ACCEPT 11,NS
        TYPE 400
400     FORMAT(' NF= ',$)
        ACCEPT 11,NF
        N=NS+NF
        CALL IFILE(20,NNAM)
        DO 500 I=1,N
        READ(20,31) (FA(I,J),J=1,N)
500     CONTINUE
        DO 1300 I=1,N
        II=I-NS
        DO 1200 J=1,N
        JJ=J-NS
        IF(II) 600,600,900
600     IF(JJ) 700,700,800
700     A(I,J)=FA(I,J)
        GO TO 1200
800     B(I,JJ)=FA(I,J)
        GO TO 1200
900     IF(JJ) 1000,1000,1100
1000    CO(II,J)=FA(I,J)
        GO TO 1200
1100    D(II,JJ)=FA(I,J)
1200    CONTINUE
1300    CONTINUE
        TYPE 1400
1400    FORMAT(' GIVE IC FILE NAME ',$)
        ACCEPT 4,NNAM
        CALL IFILE(20,NNAM)
```

```
          DO 1500 I=1,N
          READ(20,31) WIC(I)
 1500     CONTINUE
          TYPE 1600
 1600     FORMAT(' NUMBER OF ITERATIONS= ',$)
          ACCEPT 11,NITER
          TYPE 1700
 1700     FORMAT(' HSLOW= ',$)
          ACCEPT 31,HSLOW
          TYPE 1800
 1800     FORMAT(' HFAST= ',$)
          ACCEPT 31,HFAST
          TYPE 1900
 1900     FORMAT(' TMAX= ',$)
          ACCEPT 31,TMAX
          PRMT(1)=0.0
          PRMT(2)=TMAX
          PRMT(3)=HFAST
          PRMT(4)=0.001
          EWE=1.0/FLOAT(N)
          DO 2000 I=1,N
          WDOT(I)=EWE
          W(I)=WIC(I)
 2000     CONTINUE
          NSW=1
          CALL RKGS(PRMT,W,WDOT,N,IHLF,FW,OW,AUX)
          NSW=NSW-1
          DO 2200 I=1,NF
          DO 2100 J=1,NS
          AL(I,J)=0.0
 2100     CONTINUE
 2200     CONTINUE
          DO 2400 I=1,NS
          DO 2300 J=1,NF
          H(I,J)=0.0
 2300     CONTINUE
 2400     CONTINUE
          DO 6200 ITER=1,NITER
          DO 2800 I=1,NF
          DO 2700 J=1,NF
          DS(I,J)=D(I,J)
          IF(I-J) 2600,2500,2600
 2500     DI(I,I)=1.0
          GO TO 2700
 2600     DI(I,J)=0.0
 2700     CONTINUE
 2800     CONTINUE
          CALL LEQT1F(DS,NF,NF,13,DI,0,WKAREA,IER)
          DO 3200 I=1,NS
          DO 3100 J=1,NS
          SUMK=0.0
          DO 3000 K=1,NF
          SUML=0.0
          DO 2900 L=1,NF
```

```
              SUML=SUML+DI(K,L)*CO(L,J)
      2900    CONTINUE
              SUMK=SUMK+B(I,K)*SUML
      3000    CONTINUE
              A(I,J)=A(I,J)-SUMK
      3100    CONTINUE
      3200    CONTINUE
              DO 3600 I=1,NF
              DO 3500 J=1,NS
              SUMK=0.0
              DO 3400 K=1,NF
              SUML=0.0
              DO 3300 L=1,NS
              SUML=SUML+CO(K,L)*A(L,J)
      3300    CONTINUE
              SUMK=SUMK+DI(I,K)*SUML
      3400    CONTINUE
              CN(I,J)=SUMK
      3500    CONTINUE
      3600    CONTINUE
              DO 4000 I=1,NF
              DO 3900 J=1,NF
              SUMK=0.0
              DO 3800 K=1,NF
              SUML=0.0
              DO 3700 L=1,NS
              SUML=SUML+CO(K,L)*B(L,J)
      3700    CONTINUE
              SUMK=SUMK+DI(I,K)*SUML
      3800    CONTINUE
              D(I,J)=D(I,J)+SUMK
      3900    CONTINUE
      4000    CONTINUE
      4100    DO 4400 I=1,NF
              DO 4300 J=1,NS
              SUM=0.0
              DO 4200 K=1,NF
              SUM=SUM+DI(I,K)*CO(K,J)
      4200    CONTINUE
              AL(I,J)=AL(I,J)+SUM
      4300    CONTINUE
      4400    CONTINUE
      4500    DO 4700 I=1,NS
              DO 4600 J=1,NS
              AS(I,J)=A(I,J)
      4600    CONTINUE
      4700    CONTINUE
              DO 4900 I=1,NF
              DO 4800 J=1,NF
              DS(I,J)=D(I,J)
      4800    CONTINUE
      4900    CONTINUE
              CALL EIGRF(AS,NS,7,0,UA,Z,1,WKAREA,IER)
              CALL EIGRF(DS,NF,13,0,UD,Z,1,WKAREA,IER)
      DO 5100 I=1,NS
      PRINT 5000,(A(I,J),J=1,NS)
5000  FORMAT(' ',G12.5,9(1X,G12.5))
5100  CONTINUE
      PRINT 5200
5200  FORMAT(////)
      DO 5400 I=1,NS
      PRINT 5300,UA(I)
5300  FORMAT(' ',G12.5,2X,'+J',2X,G12.5)
5400  CONTINUE
      PRINT 5200
      IF(NF-10) 5500,5500,5700
5500  DO 5600 I=1,NF
      PRINT 5000,(D(I,J),J=1,NF)
5600  CONTINUE
      GO TO 6000
5700  DO 5800 I=1,NF
      PRINT 5000,(D(I,J),J=1,10)
5800  CONTINUE
      PRINT 5200
      DO 5900 I=1,NF
      PRINT 5000,(D(I,J),J=11,NF)
5900  CONTINUE
6000  PRINT 5200
      DO 6100 I=1,NF
      PRINT 5300,UD(I)
6100  CONTINUE
6110  DO 6160 I=1,NF
      DO 6130 J=1,NS
      CO(I,J)=CN(I,J)
6130  CONTINUE
6160  CONTINUE
6200  CONTINUE
      DO 6600 I=1,NF
      DO 6500 J=1,NF
      DS(I,J)=D(I,J)
      IF(I-J) 6400,6300,6400
6300  DI(I,I)=1.0
      GO TO 6500
6400  DI(I,J)=0.0
6500  CONTINUE
6600  CONTINUE
      CALL LEQT1F(DS,NF,NF,13,DI,0,WKAREA,IER)
      DO 7800 ITER=1,NITER
      DO 7000 I=1,NS
      DO 6900 J=1,NF
      SUMK=0.0
      DO 6800 K=1,NS
      SUML=0.0
      DO 6700 L=1,NF
      SUML=SUML+B(K,L)*DI(L,J)
6700  CONTINUE
      SUMK=SUMK+A(I,K)*SUML
6800  CONTINUE
```

```
          BN(I,J)=SUMK
6900      CONTINUE
7000      CONTINUE
7100      DO 7400 I=1,NS
          DU 7300 J=1,NF
          SUM=0.0
          DO 7200 K=1,NF
          SUM=SUM+B(I,K)*DI(K,J)
7200      CONTINUE
          H(I,J)=H(I.J)+SUM
7300      CONTINUE
7400      CONTINUE
7500      DU 7700 I=1,NS
          DU 7600 J=1,NF
          B(I,J)=BN(I,J)
7600      CONTINUE
7700      CONTINUE
7800      CONTINUE
          DU 9200 I=1,N
          II=I-NS
          DO 9100 J=1,N
          JJ=J-NS
          IF(II) 7900,7900,8500
7900      IF(JJ) 8000,8000,8400
8000      SUM=0.0
          DO 8100 K=1,NF
          SUM=SUM+H(I,K)*AL(K,J)
8100      CONTINUE
          IF(I-J) 8300,8200,8300
8200      TR(I,I)=1.0-SUM
          TRI(I,I)=1.0
          GO TO 9100
8300      TR(I,J)=-SUM
          TRI(I,J)=0.0
          GO TO 9100
8400      TR(I,J)=-H(I.JJ)
          TRI(I,J)=H(I.JJ)
          GO TO 9100
8500      IF(JJ) 8600,8600.8700
8600      TR(I,J)=AL(II,J)
          TRI(I,J)=-AL(II,J)
          GO TO 9100
8700      SUM=0.0
          DO 8800 K=1,NS
          SUM=SUM+AL(II,K)*H(K,JJ)
8800      CONTINUE
          IF(I-J) 9000.8900,9000
8900      TR(I,I)=1.0
          TRI(I,I)=1.0-SUM
          GO TO 9100
9000      TR(I,J)=0.0
          TRI(I,J)=-SUM
9100      CONTINUE
9200      CONTINUE
```

```
        DO 9400 I=1,NS
        XI(I)=0.0
        DO 9300 J=1,N
        XI(I)=XI(I)+TR(I,J)*WIC(J)
 9300   CONTINUE
 9400   CONTINUE
        DO 9600 I=1,NF
        II=I+NS
        ETA(I)=0.0
        DO 9500 J=1,N
        ETA(I)=ETA(I)+TR(II,J)*WIC(J)
 9500   CONTINUE
 9600   CONTINUE
        ENS=1.0/FLOAT(NS)
        DO 9700 I=1,NS
        WDOT(I)=ENS
 9700   CONTINUE
        PRMT(1)=0.0
        PRMT(2)=TMAX
        PRMT(3)=HSLOW
        PRMT(4)=0.001
        NSS=1
        CALL RKGS(PRMT,XI,WDOT,NS,IHLF,FSLO,OSLO,AUX)
        NSS=NSS-1
        ENF=1.0/FLOAT(NF)
        DO 9800 I=1,NF
        WDOT(I)=ENF
 9800   CONTINUE
        PRMT(1)=0.0
        PRMT(2)=TMAX
        PRMT(3)=HFAST
        PRMT(4)=0.001
        NSF=1
        CALL RKGS(PRMT,ETA,WDOT,NF,IHLF,FFST,OFST,AUX)
        NSF=NSF-1
        CNST=1.0/416.67
        TYPE 9900
 9900   FORMAT(' GIVE BAUD RATE ',$)
        ACCEPT 11,IBAUD
 10000  TYPE 10100
 10100  FORMAT(' WHICH STATE WILL BE PLOTTED? ',$)
        ACCEPT 11,IPLT
        IF(IPLT-21) 10140,10110,10110
 10110  IR=11+2*(IPLT-21)
        AMIN=1.0E30
        AMAX=-AMIN
        DO 10130 I=1,NSW
        PLT(I)=0.0
        DO 10120 J=1,3
        JP4=J+4
        JP7=J+7
        PLT(I)=PLT(I)+FA(IR,JP4)*WS(JP4,I)+FA(IR,JP7)*WS(JP7,I)
 10120  CONTINUE
        PLT(I)=-CNST*(PLT(I)+FA(IR,17)*WS(17,I)+FA(IR,19)*WS(19,
      1I))
      IF(AMIN.GT.PLT(I)) AMIN=PLT(I)
      IF(PLT(I).GT.AMAX) AMAX=PLT(I)
10130 CONTINUE
      GO TO 11420
10140 DO 11100 I=1,NSF
      WH(I)=0.0
      TIME=TFST(I)
      DO 10200 JT=1,NSS
      JF=JT
      IF(ABS(TIME-TSLO(JT))-1.0E-05) 10300,10300,10200
10200 CONTINUE
      GO TO 10400
10300 TINT=0.0
      GO TO 10800
10400 NM1=NSS-1
      DO 10700 JT=1,NM1
      JF=JT
      IF(TIME-TSLO(JT)) 10700,10700,10500
10500 IF(TIME-TSLO(JT+1)) 10600,10700,10700
10600 TNUM=TIME-TSLO(JT)
      TDENOM=TSLO(JT+1)-TSLO(JT)
      TINT=TNUM/TDENOM
      GO TO 10800
10700 CONTINUE
10800 DO 10900 J=1,NS
      WH(I)=WH(I)+TRI(IPLT,J)*(TINT*(XIS(J,JF+1)-XIS(J,JF))+
      1XIS(J,JF))
10900 CONTINUE
      DO 11000 J=1,NF
      JJ=J+NS
      WH(I)=WH(I)+TRI(IPLT,JJ)*ETAS(J,I)
11000 CONTINUE
11100 CONTINUE
      AMIN=1.0E30
      AMAX=-AMIN
      DO 11200 I=1,NSW
      IF(AMIN.GT.WS(IPLT,I)) AMIN=WS(IPLT,I)
      IF(WS(IPLT,I).GT.AMAX) AMAX=WS(IPLT,I)
11200 CONTINUE
      DO 11300 I=1,NSF
      IF(AMIN.GT.WH(I)) AMIN=WH(I)
      IF(WH(I).GT.AMAX) AMAX=WH(I)
11300 CONTINUE
      DO 11400 I=1,NSW
      PLT(I)=WS(IPLT,I)
11400 CONTINUE
11420 CALL INITT(IBAUD)
      CALL BINITT
      CALL DLIMY(AMIN,AMAX)
      CALL NPTS(NSW)
      CALL XFRM(4)
      CALL YFRM(4)
      CALL CHECK(TW,PLT)
```

```
        CALL DSPLAY(TW,PLT)
        CALL FRAME
        IF(IPLT-21) 11440,11460,11460
11440   CALL LINE(34)
        CALL DLIMY(AMIN,AMAX)
        CALL NPTS(NSF)
        CALL CHECK(TFST,WH)
        CALL CPLOT(TFST,WH)
11460   CALL ANMODE
        ACCEPT 1,IANS
        TYPE 11500
11500   FORMAT(' MORE PLOTS? ',S)
11600   ACCEPT 1,IANS
        IF(IANS-'Y') 11700,10000,11700
11700   IF(IANS-'N') 11800,11900,11800
11800   TYPE 40,IANS
        GO TO 11600
11900   TYPE 12000
12000   FORMAT(' RUN ANOTHER CASE? ',S)
12100   ACCEPT 1,IANS
        IF(IANS-'Y') 12200,100,12200
12200   IF(IANS-'N') 12300,12400,12300
12300   TYPE 40,IANS
        GO TO 12100
12400   STOP
        END
        !*******SUBROUTINE FW PAGE 2********
        SUBROUTINE FW(TIME,W,WDOT)
        DIMENSION W(N),WDOT(N),FA(20,20)
        COMMON/BLK1/FA,N
        DO 200 I=1,N
        WDOT(I)=0.0
        DO 100 J=1,N
        WDOT(I)=WDOT(I)+FA(I,J)*W(J)
100     CONTINUE
200     CONTINUE
        RETURN
        END
        !*******SUBROUTINE FSLO PAGE 3*******
        SUBROUTINE FSLO(TIME,XI,WDOT)
        DIMENSION XI(NS),WDOT(NS),A(7,7)
        COMMON/BLK2/A,NS
        DO 200 I=1,NS
        WDOT(I)=0.0
        DO 100 J=1,NS
        WDOT(I)=WDOT(I)+A(I,J)*XI(J)
100     CONTINUE
200     CONTINUE
        RETURN
        END
        !*******SUBROUTINE FFST PAGE 4********
        SUBROUTINE FFST(TIME,ETA,WDOT)
        DIMENSION ETA(NF),WDOT(NF),D(13,13)
        COMMON/BLK3/D,NF
```

```
          DO 200 I=1,NF
          WDOT(I)=0.0
          DO 100 J=1,NF
          WDOT(I)=WDOT(I)+D(I,J)*ETA(J)
100       CONTINUE
200       CONTINUE
          RETURN
          END
          !*******SUBROUTINE OW PAGE 5********
          SUBROUTINE OW(TIME,W,WDOT,IHLF,N,PRMT)
          DIMENSION W(N),WDOT(N),PRMT(5),WS(20,501),TW(501)
          COMMON/BLK4/WS,TW,NSW
          DO 100 I=1,N
          WS(I,NSW)=W(I)
100       CONTINUE
          TW(NSW)=TIME
          IF(NSW.GE.501) PRMT(5)=1.0
          NSW=NSW+1
          RETURN
          END
          !********SUBROUTINE OSLO PAGE 6********
          SUBROUTINE OSLO(TIME,XI,WDOT,IHLF,N,PRMT)
          DIMENSION XI(N),WDOT(N),PRMT(5),XIS(7,501),TSLO(501)
          COMMON/BLK5/XIS,TSLO,NSS
          DO 100 I=1,N
          XIS(I,NSS)=XI(I)
100       CONTINUE
          TSLO(NSS)=TIME
          IF(NSS.GE.501) PRMT(5)=1.0
          NSS=NSS+1
          RETURN
          END
          !********SUBROUTINE OFST PAGE 7********
          SUBROUTINE OFST(TIME,ETA,WDOT,IHLF,N,PRMT)
          DIMENSION ETA(N),WDOT(N),PRMT(5),ETAS(13,501),
          1TFST(501)
          COMMON/BLK6/ETAS,TFST,NSF
          DO 100 I=1,N
          ETAS(I,NSF)=ETA(I)
100       CONTINUE
          TFST(NSF)=TIME
          IF(NSF.GE.501) PRMT(5)=1.0
          NSF=NSF+1
          RETURN
          END
```

```
        !**********PROGRAM VRSMSP PAGE 1**********
        DIMENSION A(7,7),XIC(7),XEQ(7),P(7),AS(7,7),DXIC(7),WK(7)
        DIMENSION A1(2,2),A2(2,5),A3(5,2),A4(5,5),A4S(5,5),A4I
        1(5,5),A4C(5,5),DXSIC(2),DXFIC(5),A0(2,2),X(7),PRMT(5),
        2DERX(7),AUX(8,7),XBAR(2),ZBAR(5),ZWGL(5),XS(7,501),TXC
        3T(501),XBARS(2,501),ZBARS(5,501),TBAR(501),ZWGLS(5,501),
        4TWGL(501),PLT(501),CTMP(8),EFDINT(501),DELINT(501),
        5XWGLS(2,501),XC(2,501),TXC(501),C(18)
        COMPLEX U(7),Z(1,1),VT,SG,IL,US(2),UF(5)
        REAL KA,KE,KF
        COMMON/BLK8/KA,KE,KF
        COMMON/BLK9/H,D
        COMMON/BLK1/ASAT,BSAT
        COMMON/BLK2/C
        COMMON/BLK3/VREF,PIN
        COMMON/BLK4/XS,TXCT,NSXCT
        COMMON/BLK5/XBARS,ZBARS,TBAR,NSXBR
        COMMON/BLK6/ZWGLS,TWGL,NSZWGL
        COMMON/BLK7/ZBAR
        COMMON/BLK10/XC,TXC,NSXC
        EXTERNAL FXCT,FXBR,FZWGL,OXCT,OXBR,OZWGL,FZWGLC
        DATA TDOP/5.0/,TQOP/0.5/
        DATA XD/1.2/,XQ/1.0/,XP/0.25/
        DATA TA/0.06/,TE/0.5/,TF/1.0/
100     TYPE 200
200     FORMAT(' CHANGING ANY PARAMETERS?  ',S)
300     ACCEPT 400,IANS
400     FORMAT(A1)
        IF(IANS-'Y') 500,800,500
500     IF(IANS-'N') 600,6100,600
600     TYPE 700,IANS
700     FORMAT(' ??',A1,'?? TYPE AGAIN  ',S)
        GO TO 300
800     TYPE 900
900     FORMAT(' WHICH ONE?  ',S)
        ACCEPT 1000,IANS
1000    FORMAT(A4)
        IF(IANS-'H') 1100,2500,1100
1100    IF(IANS-'D') 1200,2800,1200
1200    IF(IANS-'TDOP') 1300,3000,1300
1300    IF(IANS-'TQOP') 1400,3200,1400
1400    IF(IANS-'XD') 1500,3400,1500
1500    IF(IANS-'XQ') 1600,3600,1600
1600    IF(IANS-'XP') 1700,3800,1700
1700    IF(IANS-'KA') 1800,4000,1800
1800    IF(IANS-'KE') 1900,4200,1900
1900    IF(IANS-'KF') 2000,4400,2000
2000    IF(IANS-'TA') 2100,4600,2100
2100    IF(IANS-'TE') 2200,4800,2200
2200    IF(IANS-'TF') 2230,5000,2230
2230    IF(IANS-'ASAT') 2260,5200,2260
2260    IF(IANS-'BSAT') 2300,5400,2300
2300    TYPE 2400,IANS
2400    FORMAT(' HEY DUMMY ',A4,' IS NOT ONE OF THE CHOICES'/' TRY
         1AGAIN')
         GO TO 800
2500     TYPE 2600,H
2600     FORMAT(' OLD H= ',G12.5/' NEW H= ',$)
         ACCEPT 2700,H
2700     FORMAT(G)
         GO TO 5600
2800     TYPE 2900,D
2900     FORMAT(' OLD D= ',G12.5/' NEW D= ',$)
         ACCEPT 2700,D
         GO TO 5600
3000     TYPE 3100,TDOP
3100     FORMAT(' OLD TDOP= ',G12.5/' NEW TDOP= ',$)
         ACCEPT 2700,TDOP
         GO TO 5600
3200     TYPE 3300,TQOP
3300     FORMAT(' OLD TQOP= ',G12.5/' NEW TQOP= ',$)
         ACCEPT 2700,TQOP
         GO TO 5600
3400     TYPE 3500,XD
3500     FORMAT(' OLD XD= ',G12.5/' NEW XD= ',$)
         ACCEPT 2700,XD
         GO TO 5600
3600     TYPE 3700,XQ
3700     FORMAT(' OLD XQ= ',G12.5/' NEW XQ= ',$)
         ACCEPT 2700,XQ
         GO TO 5600
3800     TYPE 3900,XP
3900     FORMAT(' OLD XP= ',G12.5/' NEW XP= ',$)
         ACCEPT 2700,XP
         GO TO 5600
4000     TYPE 4100,KA
4100     FORMAT(' OLD KA= ',G12.5/' NEW KA= ',$)
         ACCEPT 2700,KA
         GO TO 5600
4200     TYPE 4300,KE
4300     FORMAT(' OLD KE= ',G12.5/' NEW KE= ',$)
         ACCEPT 2700,KE
         GO TO 5600
4400     TYPE 4500,KF
4500     FORMAT(' OLD KF= ',G12.5/' NEW KF= ',$)
         ACCEPT 2700,KF
         GO TO 5600
4600     TYPE 4700,TA
4700     FORMAT(' OLD TA= ',G12.5/' NEW TA= ',$)
         ACCEPT 2700,TA
         GO TO 5600
4800     TYPE 4900,TE
4900     FORMAT(' OLD TE= ',G12.5/' NEW TE= ',$)
         ACCEPT 2700,TE
         GO TO 5600
5000     TYPE 5100,TF
5100     FORMAT(' OLD TF= ',G12.5/' NEW TF= ',$)
         ACCEPT 2700,TF
```

```
      GO TO 5600
5200  TYPE 5300,ASAT
5300  FORMAT(' OLD ASAT= ',G12.5/' NEW ASAT= ',S)
      ACCEPT 2700,ASAT
      GO TO 5600
5400  TYPE 5500,BSAT
5500  FORMAT(' OLD BSAT= ',G12.5/' NEW BSAT= ',S)
      ACCEPT 2700,BSAT
5600  TYPE 5700
5700  FORMAT(' CHANGING ANYMORE?  ',S)
5800  ACCEPT 400,IANS
      IF(IANS-'Y') 5900,800,5900
5900  IF(IANS-'N') 6000,6100,6000
6000  TYPE 700,IANS
      GO TO 5800
6100  TYPE 6200
6200  FORMAT(' ***ENTER LOADING INFORMATION***'/' VT=   ',S)
      ACCEPT 6300,VT
6300  FORMAT(2G)
      TYPE 6400
6400  FORMAT(' SG=   ',S)
      ACCEPT 6300,SG
      IL=CONJG(SG/VT)
      RIL=REAL(IL)
      AIIL=AIMAG(IL)
      RVT=REAL(VT)
      AIVT=AIMAG(VT)
      ANUM=XQ*AIIL-RVT
      DENOM=XQ*RIL+AIVT
      TH=ATAN2(ANUM,DENOM)
      ALPHA=ATAN2(AIVT,RVT)
      BETA=ATAN2(AIIL,RIL)
      VMAG=CABS(VT)
      AILMAG=CABS(IL)
      ANG=ALPHA-TH
      VD=VMAG*COS(ANG)
      VQ=VMAG*SIN(ANG)
      ANG=BETA-TH
      AID=AILMAG*COS(ANG)
      AIQ=AILMAG*SIN(ANG)
      XIC(1)=VQ+XP*AID
      XIC(5)=XIC(1)+(XD-XP)*AID
      XIC(2)=KF*XIC(5)/TF
      XIC(3)=TH
      XIC(4)=1.0
      XIC(6)=(KE+SE(XIC(5)))*XIC(5)
      XIC(7)=VD-XP*AIQ
      VREF=XIC(6)/KA+VMAG
      PIN=REAL(SG)
      TYPE 6500
6500  FORMAT(' GIVE FINAL ADMITTANCE'/' Y2=  ',S)
      ACCEPT 2700,Y2
      TYPE 6600
6600  FORMAT(' V2=  ',S)
```

```
        ACCEPT 2700,V2
        TYPE 6700
6700    FORMAT(' IS THERE A CHANGE IN VREF?   ',$)
6800    ACCEPT 400,IANS
        IF(IANS-'Y') 6900,7100,6900
6900    IF(IANS-'N') 7000,7300,7000
7000    TYPE 700,IANS
        GO TO 6800
7100    TYPE 7200
7200    FORMAT(' NEW VREF=   ',$)
        ACCEPT 2700,VREF
7300    DO 7400 I=1,6
        XEQ(I)=XIC(I)
7400    CONTINUE
        C(1)=1.0/TOOP
        C(2)=1.0+(XO-XP)*Y2
        C(3)=(XO-XP)*Y2*V2
        C(4)=1.0/TF
        C(5)=KF*C(4)
        C(6)=0.5/H
        C(7)=Y2*V2
        C(8)=1.0/TE
        C(9)=1.0/TA
        C(10)=(1.0-XP*Y2)*(1.0-XP*Y2)
        C(11)=(1.0-XP*Y2)*C(7)*XP
        C(12)=XP*XP*C(7)*C(7)
        C(13)=1.0/TGOP
        C(14)=(XQ-XP)*C(7)
        C(15)=1.0+(XQ-XP)*Y2
        C(16)=C(14)/C(15)
        C(17)=0.5*C(16)
        C(18)=KE+(KA*KF/TF)
        TYPE 7410
7410    FORMAT(' DO LINEAR ANALYSIS? ',$)
7420    ACCEPT 400,IANS
        IF(IANS-'Y') 7430,7450,7430
7430    IF(IANS-'N') 7440,14450,7440
7440    TYPE 700,IANS
        GO TO 7420
7450    DO 8200 ITER=1,50
        ICONV=1
        DO 7600 I=1,7
        DO 7500 J=1,7
        A(I,J)=0.0
        AS(I,J)=0.0
7500    CONTINUE
7600    CONTINUE
        VTBAR=SQRT(C(10)*(XEQ(1)*XEQ(1)+XEQ(7)*XEQ(7))+2.0*C(11)
        1*(XEQ(7)*COS(XEQ(3))-XEQ(1)*SIN(XEQ(3)))+C(12))
        P(1)=C(1)*(C(2)*XEQ(1)+C(3)*SIN(XEQ(3))-XEQ(5))
        P(2)=C(4)*(XEQ(2)-C(5)*XEQ(5))
        P(3)=377.0*(-XEQ(4)+1.0)
        P(4)=C(6)*(-PIN/XEQ(4)+C(7)*(XEQ(1)*COS(XEQ(3))+XEQ(7)*
        1SIN(XEQ(3)))+D*(XEQ(4)-1.0))
```

```
          P(5)=C(8)*((KE+SE(XEQ(5)))*XEQ(5)-XEQ(6))
          P(6)=C(9)*(KA*(-XEQ(2)+C(5)*XEQ(5)+VTBAR-VREF)+XEQ(6))
          P(7)=C(13)*(-C(14)*COS(XEQ(3))+C(15)*XEQ(7))
          A(1,1)=-C(1)*C(2)
          A(1,3)=-C(1)*C(3)*COS(XEQ(3))
          A(1,5)=C(1)
          A(2,2)=-C(4)
          A(2,5)=C(4)*C(5)
          A(3,4)=377.0
          A(4,1)=-C(6)*C(7)*COS(XEQ(3))
          A(4,3)=C(6)*C(7)*(XEQ(1)*SIN(XEQ(3))-XEQ(7)*COS(XEQ(3)))
          A(4,4)=-C(6)*(PIN/(XEQ(4)*XEQ(4))+D)
          A(4,7)=-C(6)*C(7)*SIN(XEQ(3))
          A(5,5)=-C(8)*(KE+(BSAT*XEQ(5)+1.0)*SE(XEQ(5)))
          A(5,6)=C(8)
          A(6,1)=-(C(9)*KA/VTBAR)*(C(10)*XEQ(1)-C(11)*SIN(XEQ(3)))
          A(6,2)=C(9)*KA
          A(6,3)=(C(9)*KA/VTBAR)*(C(11)*(XEQ(7)*SIN(XEQ(3))+XEQ
          1(1)*COS(XEQ(3))))
          A(6,5)=-C(9)*KA*C(5)
          A(6,6)=-C(9)
          A(6,7)=-(C(9)*KA/VTBAR)*(C(10)*XEQ(7)+C(11)*COS(XEQ(3)))
          A(7,3)=-C(13)*C(14)*SIN(XEQ(3))
          A(7,7)=-C(13)*C(15)
          DO 7800 I=1,7
          DO 7700 J=1,7
          AS(I,J)=A(I,J)
 7700     CONTINUE
 7800     CONTINUE
          CALL LEQT1F(A,1,7,7,P,0,WK,IER)
          DO 8100 I=1,7
          IF(ABS(P(I))-0.0001) 8000,8000,7900
 7900     ICONV=0
 8000     XEQ(I)=XEQ(I)+P(I)
 8100     CONTINUE
          IF(ICONV) 8200,8200,8400
 8200     CONTINUE
          TYPE 8300
 8300     FORMAT(' EQ POINT CONVERGENCE FAILURE')
          STOP
 8400     PRINT 8500
 8500     FORMAT(' ***OLD EQUILIBRIUM***')
          DO 8700 I=1,7
          PRINT 8600,I,XIC(I)
 8600     FORMAT('0',2X,'XEQ(',I1,')= ',G12.5)
 8700     CONTINUE
          PRINT 8800
 8800     FORMAT(' ***NEW EQUILIBRIUM***')
          DO 8900 I=1,7
          PRINT 8600,I,XEQ(I)
 8900     CONTINUE
          PRINT 9300
 9300     FORMAT(' ***SYSTEM A MATRIX***')
          DO 9500 I=1,7
```

```
      PRINT 9400,(AS(I,J),J=1,7)
9400  FORMAT(' ',7(2X,G12.5))
9500  CONTINUE
      DO 9600 I=1,7
      DXIC(I)=XIC(I)-XEQ(I)
9600  CONTINUE
      PRINT 9700
9700  FORMAT(' ***LINEAR INITIAL CONDITIONS***')
      DO 9900 I=1,7
      PRINT 9800,I,DXIC(I)
9800  FORMAT('0',2X,'DXIC(',I1,')= ',G12.5)
9900  CONTINUE
      DO 9960 I=1,7
      DO 9930 J=1,7
      A(I,J)=AS(I,J)
9930  CONTINUE
9960  CONTINUE
      CALL EIGRF(AS,7,7,0,U,Z,1,WK,IER)
      PRINT 10000
10000 FORMAT(' ***SYSTEM EIGENVALUES***')
      DO 10200 I=1,7
      PRINT 10100,U(I)
10100 FORMAT(' ',2X,G12.5,' +J ',G12.5)
10200 CONTINUE
      DO 10400 I=1,2
      DO 10300 J=1,2
      A1(I,J)=A(I,J)
10300 CONTINUE
10400 CONTINUE
      DO 10600 I=1,2
      DO 10500 J=1,5
      JJ=J+2
      A2(I,J)=A(I,JJ)
10500 CONTINUE
10600 CONTINUE
      DO 10800 I=1,5
      DO 10700 J=1,2
      II=I+2
      A3(I,J)=A(II,J)
10700 CONTINUE
10800 CONTINUE
      DO 11000 I=1,5
      DO 10900 J=1,5
      II=I+2
      JJ=J+2
      A4(I,J)=A(II,JJ)
      A4S(I,J)=A4(I,J)
10900 CONTINUE
11000 CONTINUE
      DO 11400 I=1,5
      DO 11300 J=1,5
      IF(I-J) 11200,11100,11200
11100 A4I(I,I)=1.0
      GO TO 11300
```

```
11200   A4I(I,J)=0.0
11300   CONTINUE
11400   CONTINUE
        CALL LEQT1F(A4,5,5,5,A4I,0,WK,IER)
        DO 11800 I=1,2
        DO 11700 J=1,2
        SUM1=0.0
        DO 11600 K=1,5
        SUM2=0.0
        DO 11500 L=1,5
        SUM2=SUM2+A2(I,L)*A4I(L,K)
11500   CONTINUE
        SUM1=SUM1+SUM2*A3(K,J)
11600   CONTINUE
        A0(I,J)=A1(I,J)-SUM1
11700   CONTINUE
11800   CONTINUE
        DO 12200 I=1,5
        DO 12100 J=1,5
        SUM1=0.0
        DO 12000 K=1,2
        SUM2=0.0
        DO 11900 L=1,5
        SUM2=SUM2+A4I(I,L)*A3(L,K)
11900   CONTINUE
        SUM1=SUM1+SUM2*A2(K,J)
12000   CONTINUE
        A4C(I,J)=A4S(I,J)+SUM1
12100   CONTINUE
12200   CONTINUE
        DO 12300 I=1,2
        DXSIC(I)=DXIC(I)
12300   CONTINUE
        DO 12600 I=1,5
        II=I+2
        SUM1=0.0
        DO 12500 K=1,2
        SUM2=0.0
        DO 12400 L=1,5
        SUM2=SUM2+A4I(I,L)*A3(L,K)
12400   CONTINUE
        SUM1=SUM1+SUM2*DXIC(K)
12500   CONTINUE
        DXFIC(I)=DXIC(II)+SUM1
12600   CONTINUE
        PRINT 12700
12700   FORMAT(' ***SLOW SUBSYSTEM MATRIX***')
        DO 12800 I=1,2
        PRINT 9400,(A0(I,J),J=1,2)
12800   CONTINUE
        CALL EIGRF(A0,2,2,0,US,Z,1,WK,IER)
        PRINT 12900
12900   FORMAT(' ***SLOW SUBSYSTEM EIGENVALUES***')
        DO 13000 I=1,2
        PRINT 10100,US(I)
13000   CONTINUE
        PRINT 13100
13100   FORMAT(' ***SLOW INITIAL CONDITIONS***')
        DO 13300 I=1,2
        PRINT 13200,I,DXSIC(I)
13200   FORMAT('0',2X,'DXSIC(',I1,')= ',G12.5)
13300   CONTINUE
        PRINT 13400
13400   FORMAT(' ***UNCORRECTED FAST SUBSYSTEM MATRIX***')
        DO 13500 I=1,5
        PRINT 9400,(A4S(I,J),J=1,5)
13500   CONTINUE
        CALL EIGRF(A4S,5,5,0,UF,Z,1,WK,IER)
        PRINT 13600
13600   FORMAT(' ***UNCORRECTED FAST EIGENVALUES***')
        DO 13700 I=1,5
        PRINT 10100,UF(I)
13700   CONTINUE
        PRINT 13800
13800   FORMAT(' ***CORRECTED FAST SUBSYSTEM MATRIX***')
        DO 13900 I=1,5
        PRINT 9400,(A4C(I,J),J=1,5)
13900   CONTINUE
        CALL EIGRF(A4C,5,5,0,UF,Z,1,WK,IER)
        PRINT 14000
14000   FORMAT(' ***CORRECTED FAST EIGENVALUES***')
        DO 14100 I=1,5
        PRINT 10100,UF(I)
14100   CONTINUE
        PRINT 14200
14200   FORMAT(' ***FAST INITIAL CONDITIONS***')
        DO 14400 I=1,5
        PRINT 14300,I,DXFIC(I)
14300   FORMAT('0',2X,'DXFIC(',I1,')= ',G12.5)
14400   CONTINUE
14450   TYPE 14500
14500   FORMAT(' ***PREPARING FOR SIMULATION***'/' FAST STEP
        1 SIZE= ',S)
        ACCEPT 2700,HFAST
        TYPE 14510
14510   FORMAT(' SLOW STEP SIZE= ',S)
        ACCEPT 2700,HSLOW
        TYPE 14520
14520   FORMAT(' TMAX= ',S)
        ACCEPT 2700,TMAX
        PRMT(4)=0.001
        TYPE 14530
14530   FORMAT(' IS THIS A FAULT? ',S)
14540   ACCEPT 400,IANS
        IF(IANS-'Y') 14550,14570,14550
14550   IF(IANS-'N') 14560,14750,14560
14560   TYPE 700,IANS
        GO TO 14540
```

```
14570   TYPE 14580
14580   FORMAT(' ***OH GOODY-A FAULT***'/' GIVE Y2 DURING
        1FAULT ',$)
        ACCEPT 2700,Y2F
        TYPE 14590
14590   FORMAT(' GIVE FAULT DURATION ',$)
        ACCEPT 2700,TCLR
        CTMP(1)=C(2)
        CTMP(2)=C(3)
        CTMP(3)=C(7)
        CTMP(4)=C(10)
        CTMP(5)=C(11)
        CTMP(6)=C(12)
        CTMP(7)=C(14)
        CTMP(8)=C(15)
        DO 14600 I=1,7
        X(I)=XIC(I)
14600   CONTINUE
        C(2)=1.0+(XD-XP)*Y2F
        C(3)=0.0
        C(7)=0.0
        C(10)=(1.0-XP*Y2F)*(1.0-XP*Y2F)
        C(11)=0.0
        C(12)=0.0
        C(14)=0.0
        C(15)=1.0+(XQ-XP)*Y2F
        PRMT(1)=0.0
        PRMT(2)=TCLR
        PRMT(3)=HFAST
        EW=1.0/7.0
        DO 14650 I=1,7
        DERX(I)=EW
14650   CONTINUE
        NSXCT=1
        CALL RKGS(PRMT,X,DERX,7,IHLF,FXCT,OXCT,AUX)
        C(2)=CTMP(1)
        C(3)=CTMP(2)
        C(7)=CTMP(3)
        C(10)=CTMP(4)
        C(11)=CTMP(5)
        C(12)=CTMP(6)
        C(14)=CTMP(7)
        C(15)=CTMP(8)
        DO 14700 I=1,7
        XIC(I)=X(I)
        TYPE *,X(I)
14700   CONTINUE
        PAUSE 'IC'
        GO TO 14850
14750   DO 14800 I=1,7
        X(I)=XIC(I)
14800   CONTINUE
14850   PRMT(1)=0.0
        PRMT(2)=TMAX
```

```
      PRMT(3)=HFAST
      EW=1.0/7.0
      DO 14900 I=1,7
      DERX(I)=EW
14900 CONTINUE
      NSXCT=1
      CALL RKGS(PRMT,X,DERX,7,IHLF,FXCT,OXCT,AUX)
      NSXCT=NSXCT-1
      PRMT(1)=0.0
      PRMT(2)=TMAX
      PRMT(3)=HSLOW
      DO 15000 I=1,2
      DERX(I)=0.5
      XBAR(I)=XIC(I)
15000 CONTINUE
      DO 15050 I=1,5
      II=I+2
      ZBAR(I)=XIC(II)
15050 CONTINUE
      NSXBR=1
      CALL RKGS(PRMT,XBAR,DERX,2,IHLF,FXBR,OXBR,AUX)
      NSXBR=NSXBR-1
      DO 15100 I=1,2
      XBAR(I)=XIC(I)
15100 CONTINUE
      DO 15150 I=1,5
      II=I+2
      ZBAR(I)=XIC(II)
15150 CONTINUE
      CALL SOLVFA(XBAR,ZBAR)
      PRMT(1)=0.0
      PRMT(2)=TMAX
      PRMT(3)=HFAST
      DO 15200 I=1,5
      II=I+2
      DERX(I)=0.2
      ZWGL(I)=XIC(II)-ZBAR(I)
15200 CONTINUE
      NSZWGL=1
      CALL RKGS(PRMT,ZWGL,DERX,5,IHLF,FZWGL,OZWGL,AUX)
      NSZWGL=NSZWGL-1
      TYPE 15300
15300 FORMAT(' ***PREPARING FOR PLOTS***'/' GIVE BAUD RATE ',
     1$)
      ACCEPT 15400,IBAUD
15400 FORMAT(I)
      TYPE 15410
15410 FORMAT(' PLOT ZERO ORDER APPROX? ',$)
15420 ACCEPT 400,IANS
      IF(IANS-'Y') 15430,15450,15430
15430 IF(IANS-'N') 15440,15460,15440
15440 TYPE 700,IANS
      GO TO 15420
15450 CALL SNGPLT(XS,TXCT,NSXCT,XBARS,ZBARS,TBAR,NSXBR,ZWGLS,
         1TWGL,NSZWGL,IBAUD,7,2,5,NSXBR,TBAR)
15400    TYPE 15500
15500    FORMAT(' DO CORRECTIONS? ',S)
15600    ACCEPT 400,IANS
         IF(IANS-'Y') 15700,15900,15700
15700    IF(IANS-'N') 15800,19400,15800
15800    TYPE 700,IANS
         GO TO 15600
15900    EFDINT(1)=0.0
         DO 16000 I=2,NSZWGL
         IM1=I-1
         EFDINT(I)=EFDINT(IM1)+0.5*(TWGL(I)-TWGL(IM1))*(ZWGLS(3,
         1I)+ZWGLS(3,IM1))
16000    CONTINUE
         EFDSUM=EFDINT(NSZWGL)
         DELINT(1)=0.0
         NSXBM1=NSXBR-1
         DO 17600 I=2,NSZWGL
         IM1=I-1
         DO 17500 J=1,2
         JT=I-J+1
         TIME=TWGL(JT)
         DO 16200 JX=1,NSXBR
         JFIND=JX
         IF(ABS(TIME-TBAR(JX))-1.0E-05) 16300,16300,16200
16200    CONTINUE
         GO TO 16600
16300    IF(J-1) 16400,16400,16500
16400    Z1BRK=ZBARS(1,JFIND)
         GO TO 17500
16500    Z1BRK1=ZBARS(1,JFIND)
         GO TO 17500
16600    DO 16900 JX=1,NSXBM1
         JFIND=JX
         IF(TIME-TBAR(JX)) 16900,16900,16700
16700    IF(TIME-TBAR(JX+1)) 16800,16900,16900
16800    TNUM=TIME-TBAR(JX)
         TDENOM=TBAR(JX+1)-TBAR(JX)
         TINT=TNUM/TDENOM
         GO TO 17200
16900    CONTINUE
         DIFF=TIME-TBAR(JFIND)
         TYPE 17000,TIME,TBAR(JFIND),DIFF
17000    FORMAT(' ',3(2X,E15.8))
         PAUSE 'CORECT'
         TYPE 17100
17100    FORMAT(' INTERPOLATION FAILURE DURING CORRECTION')
         STOP
17200    ZINT=TINT*(ZBARS(1,JFIND+1)-ZBARS(1,JFIND))+ZBARS(1,JFIN
         1D)
         IF(J-1) 17300,17300,17400
17300    Z1BRK=ZINT
         GO TO 17500
17400    Z1BRK1=ZINT
```

```
17500   CUNTINUE
        FK=SIN(Z1BRK)*(-1.0+COS(ZWGLS(1,I)))+COS(Z1BRK)*SIN(ZWG
        1LS(1,I))
        FK1=SIN(Z1BRK1)*(-1.0+COS(ZWGLS(1,IM1)))+COS(Z1BRK1)*SI
        1N(ZWGLS(1,IM1))
        DELINT(I)=DELINT(IM1)+0.5*(TWGL(I)-TWGL(IM1))*(FK+FK1)
17600   CONTINUE
        DELSUM=DELINT(NSZWGL)
        CNST1=C(1)*C(3)
        CNST2=C(4)*C(5)
        AK1=-CNST1*DELSUM+C(1)*EFDSUM
        XBAR(1)=XIC(1)+AK1
        AK2=CNST2*EFDSUM
        XBAR(2)=XIC(2)+AK2
        TYPE *,XBAR(1),XBAR(2)
        TYPE 17700
17700   FORMAT(' PRINT CORRECTION RESULTS? ',S)
17703   ACCEPT 400,IANS
        IF(IANS-'Y') 17706,17712,17706
17706   IF(IANS-'N') 17709,17790,17709
17709   TYPE 700,IANS
        GO TO 17703
17712   PRINT 17720,EFDSUM,DELSUM
17720   FORMAT(' ',G15.8,5X,G15.8)
        DU 17760 I=1,NSZWGL
        PRINT 17740,TWGL(I),ZWGLS(1,I),ZWGLS(3,I),EFDINT(I),
        1DELINT(I)
17740   FORMAT(' ',G15.8,4(5X,G15.8))
17760   CONTINUE
        DU 17780 I=1,NSXBR
        PRINT 17740,TBAR(I),ZBARS(1,I)
17780   CONTINUE
17790   PAUSE 'CORIC'
        DO 17800 I=1,NSZWGL
        XWGLS(1,I)=-AK1-CNST1*DELINT(I)+C(1)*EFDINT(I)
        XWGLS(2,I)=-AK2+CNST2*EFDINT(I)
17800   CONTINUE
        DO 17900 I=1,5
        ZBAR(I)=ZBARS(I,1)
17900   CONTINUE
        PRMT(1)=0.0
        PRMT(2)=TMAX
        PRMT(3)=HSLOW
        DO 18000 I=1,2
        DERX(I)=0.5
18000   CONTINUE
        NSXBR=1
        CALL RKGS(PRMT,XBAR,DERX,2,IHLF,FXBR,OXBR,AUX)
        NSXBR=NSXBR-1
        NSXBM1=NSXBR-1
        DO 19100 I=1,NSZWGL
        TIME=TWGL(I)
        DO 18100 JT=1,NSXBR
        JFIND=JT
        IF(ABS(TIME-TBAR(JT))-1.0E-05) 18200,18200,18100
18100   CONTINUE
        GO TO 18400
18200   DO 18300 J=1,2
        XC(J,I)=XWGLS(J,I)+XBARS(J,JFIND)
18300   CONTINUE
        GO TO 19100
18400   DO 18700 JT=1,NSXBM1
        JFIND=JT
        IF(TIME-TBAR(JT)) 18700,18700,18500
18500   IF(TIME-TBAR(JT+1)) 18600,18700,18700
18600   TNUM=TIME-TBAR(JT)
        TDENOM=TBAR(JT+1)-TBAR(JT)
        TINT=TNUM/TDENOM
        GO TO 18900
18700   CONTINUE
        DIFF=TIME-TBAR(JFIND)
        TYPE 17000,TIME,TBAR(JFIND),DIFF
        PAUSE 'XC'
        TYPE 18800
18800   FORMAT(' INTERPOLATION FAILURE AT XC')
        STOP
18900   DO 19000 J=1,2
        XBR=TINT*(XBARS(J,JFIND+1)-XBARS(J,JFIND))+XBARS(J,J
        1FIND)
        XC(J,I)=XWGLS(J,I)+XBR
19000   CONTINUE
19100   CONTINUE
        NSXC=NSZWGL
        DO 19200 I=1,NSZWGL
        TXC(I)=TWGL(I)
19200   CONTINUE
        PRMT(1)=0.0
        PRMT(2)=TMAX
        PRMT(3)=HFAST
        DO 19300 I=1,5
        DERX(I)=0.2
        ZWGL(I)=XIC(I+2)-ZBARS(I,1)
19300   CONTINUE
        NSZWGL=1
        CALL RKGS(PRMT,ZWGL,DERX,5,IHLF,FZWGLC,OZWGL,AUX)
        NSZWGL=NSZWGL-1
        CALL SNGPLT(XS,TXCT,NSXCT,XC,ZBARS,TBAR,NSXC,ZWGLS,TWGL,
        1NSZWGL,IBAUD,7,2,5,NSXBR,TXC)
        TYPE 19310
19310   FORMAT(' WANT ADDITIONAL ITERATIONS? ',$)
19320   ACCEPT 400,IANS
        IF(IANS-'Y') 19330,15900,19330
19330   IF(IANS-'N') 19340,19400,19340
19340   TYPE 700,IANS
        GO TO 19320
19400   TYPE 19500
19500   FORMAT(' RUN ANOTHER CASE? ',$)
19600   ACCEPT 400,IANS
```

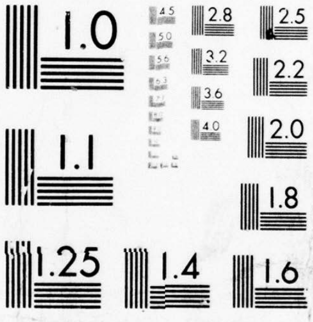
```
        IF(IANS-'Y') 19700,100,19700
19700   IF(IANS-'N') 19800,19900,19800
19800   TYPE 700,IANS
        GO TO 19600
19900   STOP
        END
!**********BLOCK DATA PAGE 2**********
        BLOCK DATA
        REAL KA,KE,KF
        COMMON/BLK1/ASAT,BSAT
        COMMON/BLK8/KA,KE,KF
        COMMON/BLK9/H,D
        DATA ASAT/0.001123/,BSAT/0.3043/
        DATA H/5.0/,D/2.0/
        DATA KA/25.0/,KE/-0.0445/,KF/0.16/
        END
!**********FUNCTION SE PAGE 3**********
        FUNCTION SE(ARG)
        COMMON/BLK1/ASAT,BSAT
        SE=ASAT*EXP(BSAT*ARG)
        RETURN
        END
!**********SUBROUTINE FXCT PAGE 4**********
        SUBROUTINE FXCT(TIME,X,DERX)
        DIMENSION X(7),DERX(7),C(18)
        REAL KA,KE,KF
        COMMON/BLK2/C
        COMMON/BLK3/VREF,PIN
        COMMON/BLK8/KA,KE,KF
        COMMON/BLK9/H,D
        CX3=COS(X(3))
        SX3=SIN(X(3))
        VLT=SQRT(C(10)*(X(1)*X(1)+X(7)*X(7))+2.0*C(11)*(X(7)*CX3
        1-X(1)*SX3)+C(12))
        DERX(1)=C(1)*(-C(2)*X(1)-C(3)*SX3+X(5))
        DERX(2)=C(4)*(-X(2)+C(5)*X(5))
        DERX(3)=377.0*(X(4)-1.0)
        DERX(4)=C(6)*(PIN/X(4)-C(7)*(X(1)*CX3+X(7)*SX3)-D*(X(4)-
        11.0))
        DERX(5)=C(8)*(-(KE+SE(X(5)))*X(5)+X(6))
        DERX(6)=C(9)*(KA*(X(2)-C(5)*X(5)-VLT+VREF)-X(6))
        DERX(7)=C(13)*(C(14)*CX3-C(15)*X(7))
        RETURN
        END
!**********SUBROUTINE FXBR PAGE 5**********
        SUBROUTINE FXBR(TIME,XBAR,DERX)
        DIMENSION XBAR(2),DERX(2),C(18),ZBAR(5)
        COMMON/BLK2/C
        COMMON/BLK7/ZBAR
        CALL SOLVFA(XBAR,ZBAR)
        DERX(1)=C(1)*(-C(2)*XBAR(1)-C(3)*SIN(ZBAR(1))+ZBAR(3))
        DERX(2)=C(4)*(-XBAR(2)+C(5)*ZBAR(3))
        RETURN
        END
```

AD-A069 859   ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB   F/G 9/3
A SINGULAR PERTURBATION APPROACH TO POWER SYSTEM DYNAMICS.(U)
AUG 78   J J ALLEMONG   DAAB07-72-C-259

UNCLASSIFIED   R-818

3 OF 3
AD A069859

END
DATE
FILMED
7 -- 79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
        !**********SUBROUTINE FZWGL PAGE 6***********
        SUBROUTINE FZWGL(TIME,ZWGL,DERX)
        DIMENSION ZWGL(5),DERX(5),C(18),XBARS(2,501),ZBARS(5,501),
        1TBAR(501),ZBAR(5),XBAR(2),Z(5)
        REAL KA,KE,KF
        COMMON/BLK2/C
        COMMON/BLK3/VREF,PIN
        COMMON/BLK5/XBARS,ZBARS,TBAR,NSXBR
        COMMON/BLK8/KA,KE,KF
        COMMON/BLK9/H,D
        DO 100 JT=1,NSXBR
        JFIND=JT
        IF(ABS(TIME-TBAR(JT))-1.0E-05) 150,150,100
100     CONTINUE
        GO TO 300
150     DO 200 I=1,2
        XBAR(I)=XBARS(I,JFIND)
200     CONTINUE
        DO 250 I=1,5
        ZBAR(I)=ZBARS(I,JFIND)
250     CONTINUE
        GO TO 750
300     NSXBM1=NSXBR-1
        DO 450 JT=1,NSXBM1
        JF=JT
        IF(TIME-TBAR(JT)) 450,450,350
350     IF(TIME-TBAR(JT+1)) 400,450,450
400     TNUM=TIME-TBAR(JT)
        TDENOM=TBAR(JT+1)-TBAR(JT)
        TINT=TNUM/TDENOM
        GO TO 600
450     CONTINUE
        DIFF=TIME-TBAR(JF)
        TYPE 500,TIME,TBAR(JF),DIFF
500     FORMAT(' ',3(2X,E15.8))
        PAUSE 'FZWGL'
        TYPE 550
550     FORMAT(' INTERPOLATION FAILURE IN FZWGL')
        STOP
600     DO 650 I=1,2
        XBAR(I)=TINT*(XBARS(I,JF+1)-XBARS(I,JF))+XBARS(I,JF)
650     CONTINUE
        DO 700 I=1,5
        ZBAR(I)=TINT*(ZBARS(I,JF+1)-ZBARS(I,JF))+ZBARS(I,JF)
700     CONTINUE
750     DO 800 I=1,5
        Z(I)=ZBAR(I)+ZWGL(I)
800     CONTINUE
        CZW1=COS(Z(1))
        SZW1=SIN(Z(1))
        VLT=SQRT(C(10)*(XBAR(1)*XBAR(1)+Z(5)*Z(5))+2.0*C(11)*(Z(5)
        1*CZW1-XBAR(1)*SZW1)+C(12))
        DERX(1)=377.0*ZWGL(2)
        DERX(2)=C(6)*(PIN/Z(2)-C(7)*(XBAR(1)*CZW1+Z(5)*SZW1)-D*
        1ZWGL(2))
        DERX(3)=C(8)*(-(KE+SE(Z(3)))*Z(3)+Z(4))
        DERX(4)=C(9)*(KA*(XBAR(2)-C(5)*Z(3)+VREF-VLT)-Z(4))
        DERX(5)=C(13)*(C(14)*CZW1-C(15)*Z(5))
        RETURN
        END
        !*********SUBROUTINE OXCT PAGE 7**********
        SUBROUTINE OXCT(TIME,X,DERX,IHLF,NDIM,PRMT)
        DIMENSION X(7),DERX(7),PRMT(5),XS(7,501),TXCT(501)
        COMMON/BLK4/XS,TXCT,NSXCT
        DO 100 I=1,7
        XS(I,NSXCT)=X(I)
100     CONTINUE
        TXCT(NSXCT)=TIME
        IF(NSXCT.GE.501) PRMT(5)=1.0
        NSXCT=NSXCT+1
        RETURN
        END
        !**********SUBROUTINE OXBR PAGE 8***********
        SUBROUTINE OXBR(TIME,XBAR,DERX,IHLF,NDIM,PRMT)
        DIMENSION XBAR(2),DERX(2),PRMT(5),XBARS(2,501),TBAR(501),
        1ZBAR(5),ZBARS(5,501)
        COMMON/BLK5/XBARS,ZBARS,TBAR,NSXBR
        COMMON/BLK7/ZBAR
        CALL SOLVFA(XBAR,ZBAR)
        DO 100 I=1,2
        XBARS(I,NSXBR)=XBAR(I)
100     CONTINUE
        DO 200 I=1,5
        ZBARS(I,NSXBR)=ZBAR(I)
200     CONTINUE
        TBAR(NSXBR)=TIME
        IF(NSXBR.GE.501) PRMT(5)=1.0
        NSXBR=NSXBR+1
        RETURN
        END
        !**********SUBROUTINE OZWGL PAGE 9**********
        SUBROUTINE OZWGL(TIME,ZWGL,DERX,IHLF,NDIM,PRMT)
        DIMENSION ZWGL(5),DERX(5),PRMT(5),ZWGLS(5,501),TWGL(501)
        COMMON/BLK6/ZWGLS,TWGL,NSZWGL
        DO 100 I=1,5
        ZWGLS(I,NSZWGL)=ZWGL(I)
100     CONTINUE
        TWGL(NSZWGL)=TIME
        IF(NSZWGL.GE.501) PRMT(5)=1.0
        NSZWGL=NSZWGL+1
        RETURN
        END
        !**********SUBROUTINE SOLVFA PAGE 10***********
        SUBROUTINE SOLVFA(XBAR,ZBAR)
        DIMENSION XBAR(2),ZBAR(5),C(18)
        REAL KA,KE,KF
        COMMON/BLK2/C
        COMMON/BLK3/VREF,PIN
        COMMON/BLK1/ASAT,BSAT
        COMMON/BLK8/KA,KE,KF
        COMMON/BLK9/H,D
        DIT=ZBAR(1)
        DO 100 ITER=1,50
        ANUM=-PIN+C(7)*(XBAR(1)*COS(DIT)+C(17)*SIN(2.0*DIT))
        DENOM=C(7)*(XBAR(1)*SIN(DIT)-C(16)*COS(2.0*DIT))
        DEL=ANUM/DENOM
        DIT=DIT+DEL
        IF(ABS(DEL)-0.0001) 300,300,100
100     CONTINUE
        TYPE 200
200     FORMAT(' CONVERGENCE FAILURE ON DELTA-SOLVFA')
        STOP
300     ZBAR(1)=DIT
        ZBAR(5)=C(16)*COS(ZBAR(1))
        ZBAR(2)=1.0
        VTBAR=SQRT(C(10)*(XBAR(1)*XBAR(1)+ZBAR(5)*ZBAR(5))+2.0*C(
       111)*(ZBAR(5)*COS(ZBAR(1))-XBAR(1)*SIN(ZBAR(1)))+C(12))
        DIT=ZBAR(3)
        DO 400 ITER=1,50
        ANUM=-(C(18)+SE(DIT))*DIT+KA*(XBAR(2)+VREF-VTBAR)
        DENOM=C(18)+(BSAT*DIT+1.0)
        DEL=ANUM/DENOM
        DIT=DIT+DEL
        IF(ABS(DEL)-0.0001) 600,600,400
400     CONTINUE
        TYPE 500
500     FORMAT(' CONVERGENCE FAILURE ON EFD-SOLVFA')
        STOP
600     ZBAR(3)=DIT
        ZBAR(4)=KA*(XBAR(2)-C(5)*ZBAR(3)+VREF-VTBAR)
        RETURN
        END
        !***********SUBROUTINE FZWGLC PAGE 11**********
        SUBROUTINE FZWGLC(TIME,ZWGL,DERX)
        DIMENSION ZWGL(5),DERX(5),C(18),XBARS(2,501),ZBARS(5,501
       1),TBAR(501),XC(2,501),TXC(501),ZBAR(5),X(2),Z(5)
        REAL KA,KE,KF
        COMMON/BLK2/C
        COMMON/BLK3/VREF,PIN
        COMMON/BLK5/XBARS,ZBARS,TBAR,NSXBR
        COMMON/BLK8/KA,KE,KF
        COMMON/BLK9/H,D
        COMMON/BLK10/XC,TXC,NSXC
        DO 100 JT=1,NSXC
        JFIND=JT
        IF(ABS(TIME-TXC(JT))-1.0E-05) 200,200,100
100     CONTINUE
        GO TO 400
200     DO 300 I=1,2
        X(I)=XC(I,JFIND)
300     CONTINUE
        GO TO 1200
```

```
400      NSXCM1=NSXC-1
         DO 700 JT=1,NSXCM1
         JFIND=JT
         IF(TIME-TXC(JT)) 700,700,500
500      IF(TIME-TXC(JT+1)) 600,700,700
600      TNUM=TIME-TXC(JT)
         TDENOM=TXC(JT+1)-TXC(JT)
         TINT=TNUM/TDENOM
         GO TO 1000
700      CONTINUE
         DIFF=TIME-TXC(JFIND)
         TYPE 800,TIME,TXC(JFIND),DIFF
800      FORMAT(' ',3(2X,E15.8))
         PAUSE 'FZWGLC'
         TYPE 900
900      FORMAT(' INTERPOLATION FAILURE IN FZWGLC-XC')
         STOP
1000     DO 1100 I=1,2
         X(I)=TINT*(XC(I,JFIND+1)-XC(I,JFIND))+XC(I,JFIND)
1100     CONTINUE
1200     DO 1300 JT=1,NSXBR
         JFIND=JT
         IF(ABS(TIME-TBAR(JT))-1.0E-05) 1400,1400,1300
1300     CONTINUE
         GO TO 1600
1400     DO 1500 I=1,5
         ZBAR(I)=ZBARS(I,JFIND)
1500     CONTINUE
         GO TO 2300
1600     NSXBM1=NSXBR-1
         DO 1900 JT=1,NSXBM1
         JFIND=JT
         IF(TIME-TBAR(JT)) 1900,1900,1700
1700     IF(TIME-TBAR(JT+1)) 1800,1900,1900
1800     TNUM=TIME-TBAR(JT)
         TDENOM=TBAR(JT+1)-TBAR(JT)
         TINT=TNUM/TDENOM
         GO TO 2100
1900     CONTINUE
         DIFF=TIME-TBAR(JFIND)
         TYPE 800,TIME,TBAR(JFIND),DIFF
         PAUSE 'FZWGLC'
         TYPE 2000
2000     FORMAT(' INTERPOLATION FAILURE IN FZWGLC-ZBAR')
         STOP
2100     DO 2200 I=1,5
         ZBAR(I)=TINT*(ZBARS(I,JFIND+1)-ZBARS(I,JFIND))+ZBARS
         1(I,JFIND)
2200     CONTINUE
2300     DO 2400 I=1,5
         Z(I)=ZBAR(I)+ZWGL(I)
2400     CONTINUE
         CZW1=COS(Z(1))
         SZW1=SIN(Z(1))
```

```
      VLT=SQRT(C(10)*(X(1)*X(1)+Z(5)*Z(5))+2.0*C(11)*(Z(5)*
     1CZW1-X(1)*SZW1)+C(12))
      DERX(1)=377.0*ZWGL(2)
      DERX(2)=C(6)*(PIN/Z(2)-C(7)*(X(1)*CZW1+Z(5)*SZW1)-
     1D*ZWGL(2))
      DERX(3)=C(8)*(-(KE+SE(Z(3)))*Z(3)+Z(4))
      DERX(4)=C(9)*(KA*(-X(2)-C(5)*Z(3)+VREF-VLT)-Z(4))
      DERX(5)=C(13)*(C(14)*CZW1-C(15)*Z(5))
      RETURN
      END
      !***********SUBROUTINE SNGPLT PAGE 12**********
      SUBROUTINE SNGPLT(ES,TX,NSTPE,XBARS,ZBARS,TBAR,NSTPXB,ZWS,
     1TW,NSTPZW,IBAUD,N,NS,NF,NSTPZB,TXC)
      DIMENSION ES(N,NSTPE),TX(NSTPE),XBARS(NS,NSTPXB),ZBARS(NF,
     1NSTPZB),ZWS(NF,NSTPZW),TBAR(NSTPZB),TW(NSTPZW),PLT(501),
     2ZHAT(501),TXC(NSTPXB)
400   FORMAT(A1)
      DO 6400 IPLT=1,NS
      AMIN=1.0E30
      AMAX=-1.0E30
      DO 6000 JT=1,NSTPE
      IF(AMIN.GT.ES(IPLT,JT)) AMIN=ES(IPLT,JT)
      IF(ES(IPLT,JT).GT.AMAX) AMAX=ES(IPLT,JT)
6000  CONTINUE
      DO 6100 JT=1,NSTPXB
      IF(AMIN.GT.XBARS(IPLT,JT)) AMIN=XBARS(IPLT,JT)
      IF(XBARS(IPLT,JT).GT.AMAX) AMAX=XBARS(IPLT,JT)
6100  CONTINUE
      DO 6200 JT=1,NSTPE
      PLT(JT)=ES(IPLT,JT)
6200  CONTINUE
      CALL INITT(IBAUD)
      CALL BINITT
      CALL DLIMY(AMIN,AMAX)
      CALL NPTS(NSTPE)
      CALL XFRM(4)
      CALL YFRM(4)
      CALL CHECK(TX,PLT)
      CALL DSPLAY(TX,PLT)
      CALL FRAME
      DO 6300 JT=1,NSTPXB
      PLT(JT)=XBARS(IPLT,JT)
6300  CONTINUE
      CALL LINE(34)
      CALL DLIMY(AMIN,AMAX)
      CALL NPTS(NSTPXB)
      CALL CHECK(TXC,PLT)
      CALL CPLOT(TXC,PLT)
      CALL ANMODE
      ACCEPT 400,IANS
6400  CONTINUE
      DO 7600 IZHAT=1,NF
      ZHAT(1)=ZBARS(IZHAT,1)+ZWS(IZHAT,1)
      DO 6800 ITZW=2,NSTPZW
```

```
          DO 6500 ITX8=2,NSTPZ8
          NTX8=ITX8
          IF(TW(ITZW).LE.TBAR(ITX8)) GO TO 6700
6500      CONTINUE
          IF((TW(ITZW)-TBAR(NTX8))-1.0E-04) 6700,6700,6550
6550      TYPE 6600
6600      FORMAT(' ***SEARCH FAILURE DURING INTERPOLATION***')
          STOP
6700      ZINT=((TW(ITZW)-TBAR(NTX8-1))*(ZBARS(IZHAT,NTX8)-ZBARS
          1(IZHAT,NTX8-1))/(TBAR(NTX8)-TBAR(NTX8-1)))+ZBARS(IZHAT,
          2NTX8-1)
          ZHAT(ITZW)=ZWS(IZHAT,ITZW)+ZINT
6800      CONTINUE
          IZHAT2=IZHAT+NS
          AMIN=1.0E30
          AMAX=-1.0E30
          DO 6900 JT=1,NSTPE
          IF(AMIN.GT.ES(IZHAT2,JT)) AMIN=ES(IZHAT2,JT)
          IF(ES(IZHAT2,JT).GT.AMAX)AMAX=ES(IZHAT2,JT)
6900      CONTINUE
          DO 7000 JT=1,NSTPZW
          IF(AMIN.GT.ZHAT(JT)) AMIN=ZHAT(JT)
          IF(ZHAT(JT).GT.AMAX) AMAX=ZHAT(JT)
7000      CONTINUE
          DO 7100 JT=1,NSTPE
          PLT(JT)=ES(IZHAT2,JT)
7100      CONTINUE
          CALL INITT(IBAUD)
          CALL BINITT
          CALL DLIMY(AMIN,AMAX)
          CALL NPTS(NSTPE)
          CALL XFRM(4)
          CALL YFRM(4)
          CALL CHECK(TX,PLT)
          CALL DSPLAY(TX,PLT)
          CALL FRAME
          CALL LINE(34)
          CALL DLIMY(AMIN,AMAX)
          CALL NPTS(NSTPZW)
          CALL CHECK(TW,ZHAT)
          CALL CPLOT(TW,ZHAT)
          CALL ANMODE
          ACCEPT 400,IANS
          AMIN=1.0E30
          AMAX=-1.0E30
          DO 7200 JT=1,NSTPZ8
          IF(AMIN.GT.ZBARS(IZHAT,JT)) AMIN=ZBARS(IZHAT,JT)
          IF(ZBARS(IZHAT,JT).GT.AMAX) AMAX=ZBARS(IZHAT,JT)
7200      CONTINUE
          DO 7300 JT=1,NSTPZW
          IF(AMIN.GT.ZWS(IZHAT,JT)) AMIN=ZWS(IZHAT,JT)
          IF(ZWS(IZHAT,JT).GT.AMAX) AMAX=ZWS(IZHAT,JT)
7300      CONTINUE
          DO 7400 JT=1,NSTPZ8
          PLT(JT)=ZBARS(IZHAT,JT)
 7400     CONTINUE
          CALL INITT(IBAUD)
          CALL BINITT
          CALL DLIMY(AMIN,AMAX)
          CALL NPTS(NSTPZB)
          CALL XFRM(4)
          CALL YFRM(4)
          CALL CHECK(TBAR,PLT)
          CALL DSPLAY(TBAR,PLT)
          CALL FRAME
          DO 7500 JT=1,NSTPZW
          PLT(JT)=ZWS(IZHAT,JT)
 7500     CONTINUE
          CALL LINE(34)
          CALL DLIMY(AMIN,AMAX)
          CALL NPTS(NSTPZW)
          CALL CHECK(TW,PLT)
          CALL CPLOT(TW,PLT)
          CALL ANMODE
          ACCEPT 400,IANS
 7600     CONTINUE
          RETURN
          END
          !**********PROGRAM MMSP PAGE 1***********
          DIMENSION H(3),DEL(3),XQ(3),VD(3),VQ(3),XIC(2Ø),XP(3),
          1XD(3),TF(3),VREF(3),PIN(3),Y(3,3),TH(3,3),PRMT(5),
          2XDOT(2Ø),X(2Ø),AUX(8,2Ø),XS(23,5Ø1),PLT(5Ø1),TS(5Ø1)
          3,D(3),TA(3),TE(3),TDOP(3),TQOP(3),VLT(3),XBAR(7),
          4ZBAR(13),XBARS(7,5Ø1),ZBARS(13,5Ø1),TBAR(5Ø1),XC(7,
          55Ø1),TXC(5Ø1),ZWGLS(13,5Ø1),TWGL(5Ø1),ZWGL(13),DELRW(3),
          6AINT1(5Ø1),AINT2(5Ø1),AINT3(5Ø1),AINT4(5Ø1),XWGLS(7,
          75Ø1),AINT5(5Ø1),DELR(3),ZDOT(13,5Ø1),TDOT(5Ø1)
          COMPLEX ITC,YF(3,3),YPD(3,3),SG(3),VT(3)
          REAL IMAG,ID(3),IQ(3),KF(3),KE(3),KA(3),IDB(3,5Ø1),IQB(
          13,5Ø1),IDW(3,5Ø1),IQW(3,5Ø1)
          COMMON/BLK1/XD,XQ,XP
          COMMON/BLK2/TDOP,TQOP
          COMMON/BLK3/TA,TE,TF
          COMMON/BLK4/H,D
          COMMON/BLK5/ASAT,BSAT
          COMMON/BLK6/KA,KE,KF
          COMMON/BLK7/Y,TH
          COMMON/BLK8/BH,SH
          COMMON/BLK9/VREF,PIN
          COMMON/BLK1Ø/XS,TS,NXCT
          COMMON/BLK11/VLT
          COMMON/BLK12/ZBAR
          COMMON/BLK13/XBARS,ZBARS,TBAR,NXBR
          COMMON/BLK14/XC,TXC,NXC
          COMMON/BLK15/ICRCT
          COMMON/BLK16/ZWGLS,TWGL,NZWGL
          COMMON/BLK17/ZDOT,TDOT,NDOT
          EXTERNAL FXCT,OXCT,FXBAR,OXBAR,FZWGL,OZWGL
          DATA (SG(I),I=1,3)/(Ø.723,Ø.27Ø3),(1.63,Ø.Ø654),(Ø.85,
          1-Ø.1Ø95)/
          DATA (VT(I),I=1,3)/(1.Ø4,Ø.Ø),(1.Ø12,Ø.1648),(1.Ø22,Ø.Ø
          18292)/
1         FORMAT(A1)
4         FORMAT(A4)
11        FORMAT(I)
31        FORMAT(G)
32        FORMAT(2G)
          SH=H(2)+H(3)
          BH=H(1)+SH
1ØØ       DO 2ØØ I=1.3
          ITC=CONJG(SG(I)/VT(I))
          IMAG=CABS(ITC)
          BETA=ATAN2(AIMAG(ITC),REAL(ITC))
          VMAG=CABS(VT(I))
          ALPHA=ATAN2(AIMAG(VT(I)),REAL(VT(I)))
          ANUM=XQ(I)*AIMAG(ITC)-REAL(VT(I))
          DENOM=XQ(I)*REAL(ITC)+AIMAG(VT(I))
          DEL(I)=ATAN2(ANUM,DENOM)
          ANG=ALPHA-DEL(I)
          VD(I)=VMAG*COS(ANG)
          VQ(I)=VMAG*SIN(ANG)
          ANG=BETA-DEL(I)
```

```
        ID(I)=IMAG*COS(ANG)
        IQ(I)=IMAG*SIN(ANG)
200     CONTINUE
        XIC(1)=1.0
        DO 300 I=1,3
        XIC(I+1)=VQ(I)+XP(I)*ID(I)
300     CONTINUE
        DO 400 I=1,3
        XIC(I+11)=VQ(I)+XD(I)*ID(I)
400     CONTINUE
        DO 500 I=1,3
        XIC(I+4)=KF(I)*XIC(I+11)/TF(I)
500     CONTINUE
        DO 600 I=1,3
        XIC(I+14)=(KE(I)+SE(XIC(I+11)))*XIC(I+11)
600     CONTINUE
        DO 700 I=1,3
        XIC(I+17)=VD(I)-XP(I)*IQ(I)
700     CONTINUE
        DO 800 I=1,3
        VREF(I)=(XIC(I+14)/KA(I))+CABS(VT(I))
        PIN(I)=REAL(SG(I))
800     CONTINUE
        XIC(8)=(H(1)/BH)*((H(2)*DEL(2)+H(3)*DEL(3))/SH-DEL(1))
        XIC(9)=0.0
        XIC(10)=DEL(2)-DEL(3)
        XIC(11)=0.0
        TYPE 900
900     FORMAT(' GIVE STUDY DURATION ',S)
        ACCEPT 31,TMAX
        TYPE 1000
1000    FORMAT(' GIVE SLOW STEP SIZE ',S)
        ACCEPT 31,HSLOW
        TYPE 1100
1100    FORMAT(' GIVE FAST STEP SIZE ',S)
        ACCEPT 31,HFAST
        TYPE 1200
1200    FORMAT(' IS THIS A FAULT? ',S)
1300    ACCEPT 1,IANS
        IF(IANS-'Y') 1400,1700,1400
1400    IF(IANS-'N') 1500,2600,1500
1500    TYPE 1600,IANS
1600    FORMAT(' ??',A1,'?? TYPE AGAIN ',S)
        GO TO 1300
1700    TYPE 1800
1800    FORMAT(' GIVE CLEARING TIME ',S)
        ACCEPT 31,TCLEAR
        TYPE 1900
1900    FORMAT(' GIVE NAME OF FAULTED Y MATRIX ',S)
        ACCEPT 4,NNAM
        CALL IFILE(20,NNAM)
        DO 2000 I=1,3
        READ(20,32) (YF(I,J),J=1,3)
2000    CONTINUE
```

```
      DO 2300 I=1,3
      DO 2200 J=I,3
      Y(I,J)=CABS(YF(I,J))
      TH(I,J)=ATAN2(AIMAG(YF(I,J)),REAL(YF(I,J)))
      IF(I-J) 2100,2200,2100
2100  Y(J,I)=Y(I,J)
      TH(J,I)=TH(I,J)
2200  CONTINUE
2300  CONTINUE
      PRMT(1)=0.0
      PRMT(2)=TCLEAR
      PRMT(3)=HFAST
      PRMT(4)=0.001
      DO 2400 I=1,20
      XDOT(I)=0.05
      X(I)=XIC(I)
2400  CONTINUE
      NXCT=1
      CALL RKGS(PRMT,X,XDOT,20,IHLF,FXCT,OXCT,AUX)
      NXCT=NXCT-1
      DO 2500 I=1,20
      XIC(I)=XS(I,NXCT)
      TYPE *,XS(I,NXCT)
2500  CONTINUE
      PAUSE 'IC'
2600  PRMT(1)=0.0
      PRMT(2)=TMAX
      PRMT(3)=HFAST
      PRMT(4)=0.001
      DO 2700 I=1,20
      XDOT(I)=0.05
      X(I)=XIC(I)
2700  CONTINUE
      TYPE 2800
2800  FORMAT(' GIVE FILE NAME OF POST-DISTURBANCE Y
      1MATRIX ',S)
      ACCEPT 4,NNAM
      CALL IFILE(20,NNAM)
      DO 2900 I=1,3
      READ(20,32) (YPD(I,J),J=1,3)
2900  CONTINUE
      DO 3200 I=1,3
      DO 3100 J=I,3
      Y(I,J)=CABS(YPD(I,J))
      TH(I,J)=ATAN2(AIMAG(YPD(I,J)),REAL(YPD(I,J)))
      IF(I-J) 3000,3100,3000
3000  Y(J,I)=Y(I,J)
      TH(J,I)=TH(I,J)
3100  CONTINUE
3200  CONTINUE
      NXCT=1
      CALL RKGS(PRMT,X,XDOT,20,IHLF,FXCT,OXCT,AUX)
      NXCT=NXCT-1
      DO 3300 I=1,7
      XBAR(I)=XIC(I)
3300  CONTINUE
      DO 3400 I=1,13
      ZBAR(I)=XIC(I+7)
3400  CONTINUE
      PRMT(1)=0.0
      PRMT(2)=TMAX
      PRMT(3)=HSLOW
      PRMT(4)=0.001
      EWS=1.0/7.0
      DO 3500 I=1,7
      XDOT(I)=EWS
3500  CONTINUE
      NXBR=1
      CALL RKGS(PRMT,XBAR,XDOT,7,IHLF,FXBAR,OXBAR,AUX)
      NXBR=NXBR-1
      PRMT(1)=0.0
      PRMT(2)=TMAX
      PRMT(3)=HFAST
      PRMT(4)=0.001
      EWF=1.0/13.0
      DO 3600 I=1,13
      XDOT(I)=EWF
      ZWGL(I)=XIC(I+7)-ZBARS(I,1)
3600  CONTINUE
      NZWGL=1
      ICRCT=0
      CALL RKGS(PRMT,ZWGL,XDOT,13,IHLF,FZWGL,OZWGL,AUX)
      NZWGL=NZWGL-1
      TYPE 3700
3700  FORMAT(' GIVE BAUD RATE ',$)
      ACCEPT 11,IBAUD
      TYPE 3800
3800  FORMAT(' PLOT ZERO ORDER RESULTS? ',$)
3900  ACCEPT 1,IANS
      IF(IANS-'Y') 4000,4200,4000
4000  IF(IANS-'N') 4100,4300,4100
4100  TYPE 1600,IANS
      GO TO 3900
4200  CALL GRAPHS(NXCT,NXBR,NXBR,NZWGL,IBAUD,XS,TS,XBARS,
      1TBAR,ZBARS,TBAR,ZWGLS,TWGL)
4300  TYPE 4400
4400  FORMAT(' DO CORRECTIONS? ',$)
4500  ACCEPT 1,IANS
      IF(IANS-'Y') 4600,4800,4600
4600  IF(IANS-'N') 4700,8700,4700
4700  TYPE 1600,IANS
      GO TO 4500
4800  DELR(1)=0.0
      DELRW(1)=0.0
      DO 5400 I=1,3
      DO 5000 IT=1,NXBR
      IDB(I,IT)=0.0
      IQB(I,IT)=0.0
      DELR(2)=BH*ZBARS(1,IT)/H(1)+H(3)*ZBARS(3,IT)/SH
      DELR(3)=BH*ZBARS(1,IT)/H(1)-H(2)*ZBARS(3,IT)/SH
      DO 4900 J=1,3
      ANG=TH(I,J)+DELR(J)-DELR(I)
      CANG=COS(ANG)
      SANG=SIN(ANG)
      IOB(I,IT)=IOB(I,IT)+Y(I,J)*(ZBARS(J+10,IT)*CANG-XBARS(
     1J+1,IT)*SANG)
      IQB(I,IT)=IQB(I,IT)+Y(I,J)*(XBARS(J+1,IT)*CANG+ZBARS(
     1J+10,IT)*SANG)
4900  CONTINUE
5000  CONTINUE
      DO 5300 IT=1,NZWGL
      IOW(I,IT)=0.0
      IQW(I,IT)=0.0
      TIME=TWGL(IT)
      CALL INTPOL(TIME,TBAR,NXBR,JF,TINT)
      DELC=TINT*(ZBARS(1,JF+1)-ZBARS(1,JF))+ZBARS(1,JF)
      DELD=TINT*(ZBARS(3,JF+1)-ZBARS(3,JF))+ZBARS(3,JF)
      DELR(2)=BH*DELC/H(1)+H(3)*DELD/SH
      DELR(3)=BH*DELC/H(1)-H(2)*DELD/SH
      DELRW(2)=BH*ZWGLS(1,IT)/H(1)+H(3)*ZWGLS(3,IT)/SH
      DELRW(3)=BH*ZWGLS(1,IT)/H(1)-H(2)*ZWGLS(3,IT)/SH
      DO 5200 J=1,3
5100  EQP=TINT*(XBARS(J+1,JF+1)-XBARS(J+1,JF))+XBARS(J+1,
     1JF)
      EDP=TINT*(ZBARS(J+10,JF+1)-ZBARS(J+10,JF))+ZBARS(
     1J+10,JF)+ZWGLS(J+10,IT)
      EDPS=EDP-ZWGLS(J+10,IT)
      ANGB=TH(I,J)+DELR(J)-DELR(I)
      CANG=COS(ANGB)
      SANG=SIN(ANGB)
      ANGW=DELRW(J)-DELRW(I)
      T1=-1.0+COS(ANGW)
      T2=SIN(ANGW)
      T3=T1+1.0
      IOW(I,IT)=IOW(I,IT)+Y(I,J)*((EDPS*CANG-EQP*SANG)*T1-
     1(EDP*SANG+EQP*CANG)*T2+ZWGLS(J+10,IT)*CANG*T3)
      IQW(I,IT)=IQW(I,IT)+Y(I,J)*((EQP*CANG+EDPS*SANG)*T1+
     1(EDP*CANG-EQP*SANG)*T2+ZWGLS(J+10,IT)*SANG*T3)
5200  CONTINUE
5300  CONTINUE
5400  CONTINUE
      AINT1(1)=0.0
      AINT2(1)=0.0
      AINT3(1)=0.0
      AINT4(1)=0.0
      DO 5800 IT=2,NZWGL
      IF(IT-2) 5500,5500,5600
5500  F1M1=ZWGLS(2,1)/(XBARS(1,1)*XBARS(1,1))
      F2M1=ZWGLS(4,1)/(XBARS(1,1)*XBARS(1,1))
      GO TO 5700
5600  F1M1=F1
      F2M1=F2
```

```
5700      IM1=IT-1
          CALL INTPOL(TWGL(IT),TBAR,NXBR,JF,TINT)
          OMR=TINT*(XBARS(1,JF+1)-XBARS(1,JF))+XBARS(1,JF)
          OMRS=OMR*OMR
          F1=ZWGLS(2,IT)/OMRS
          F2=ZWGLS(4,IT)/OMRS
          DT=TWGL(IT)-TWGL(IM1)
          AINT1(IT)=AINT1(IM1)+0.5*DT*(F1+F1M1)
          AINT2(IT)=AINT2(IM1)+0.5*DT*(F2+F2M1)
          AINT3(IT)=AINT3(IM1)+0.5*DT*(ZWGLS(2,IT)+ZWGLS(2,IM1))
          AINT4(IT)=AINT4(IM1)+0.5*DT*(ZWGLS(4,IT)+ZWGLS(4,IM1))
5800      CONTINUE
          AINT5(1)=0.0
          DO 6400 IT=2,NZWGL
          IM1=IT-1
          CALL INTPOL(TWGL(IT),TBAR,NXBR,JF,TINT)
          IF(IT-2) 5900,5900,6100
5900      F1M1=0.0
          DO 6000 J=1,3
          F1M1=F1M1+XBARS(J+1,1)*IQW(J,1)+(ZBARS(J+10,1)+ZWGLS(J
          1+10,1))*IDW(J,1)+ZWGLS(J+10,1)*IDB(J,1)
6000      CONTINUE
          GO TO 6200
6100      F1M1=F1
6200      F1=0.0
          DO 6300 J=1,3
          EQP=TINT*(XBARS(J+1,JF+1)-XBARS(J+1,JF))+XBARS(J+1,JF)
          EDP=TINT*(ZBARS(J+10,JF+1)-ZBARS(J+10,JF))+ZBARS(J+10,
          1JF)+ZWGLS(J+10,IT)
          AIDB=TINT*(IDB(J,JF+1)-IDB(J,JF))+IDB(J,JF)
          F1=F1+EQP*IQW(J,IT)+EDP*IDW(J,IT)+ZWGLS(J+10,IT)*AIDB
6300      CONTINUE
          DT=TWGL(IT)-TWGL(IM1)
          AINT5(IT)=AINT5(IM1)+0.5*DT*(F1+F1M1)
6400      CONTINUE
          C1=(0.5*SH*PIN(1)/(H(1)*BH))*AINT1(NZWGL)
          C2=(0.5*PIN(2)/BH)*(-AINT1(NZWGL)-H(3)*AINT2(NZWGL)
          1/SH)
          C3=(0.5*PIN(3)/BH)*(-AINT1(NZWGL)+H(2)*AINT2(NZWGL)
          1/SH)
          C4=0.5*D(1)*SH*AINT3(NZWGL)/(H(1)*BH)
          C5=(0.5*D(2)/BH)*(-AINT3(NZWGL)-H(3)*AINT4(NZWGL)/
          1SH)
          C6=(0.5*D(3)/BH)*(-AINT3(NZWGL)+H(2)*AINT4(NZWGL)/
          1SH)
          C7=-(0.5/BH)*AINT5(NZWGL)
          XBAR(1)=XIC(1)+C1+C2+C3+C4+C5+C6+C7
          DO 6500 IT=1,NZWGL
          T1=-C1+(0.5*SH*PIN(1)/(H(1)*BH))*AINT1(IT)
          T2=-C2+(0.5*PIN(2)/BH)*(-AINT1(IT)-H(3)*AINT2(IT)/
          1SH)
          T3=-C3+(0.5*PIN(3)/BH)*(-AINT1(IT)+H(2)*AINT2(IT)/
          1SH)
          T4=-C4+0.5*D(1)*SH*AINT3(IT)/(H(1)*BH)
```

```
      T5=-C5+(0.5*D(2)/BH)*(-AINT3(IT)-H(3)*AINT4(IT)/
     1SH)
      T6=-C6+(0.5*D(3)/BH)*(-AINT3(IT)+H(2)*AINT4(IT)/
     1SH)
      T7=-C7-(0.5/BH)*AINT5(IT)
      XWGLS(1,IT)=T1+T2+T3+T4+T5+T6+T7
6500  CONTINUE
      DO 6900 I=1,3
      AINT1(1)=0.0
      AINT2(1)=0.0
      DO 6600 IT=2,NZWGL
      IM1=IT-1
      DT=TWGL(IT)-TWGL(IM1)
      AINT1(IT)=AINT1(IM1)+0.5*DT*(IDW(I,IT)+IDW(I,IM1))
      AINT2(IT)=AINT2(IM1)+0.5*DT*(ZWGLS(I+4,IT)+ZWGLS(
     1I+4,IM1))
6600  CONTINUE
      C1=-((XD(I)-XP(I))/TDOP(I))*AINT1(NZWGL)
      C2=(1.0/TDOP(I))*AINT2(NZWGL)
      XBAR(I+1)=XIC(I+1)+C1+C2
      DO 6700 IT=1,NZWGL
      T1=-C1-((XD(I)-XP(I))/TDOP(I))*AINT1(IT)
      T2=-C2+(1.0/TDOP(I))*AINT2(IT)
      XWGLS(I+1,IT)=T1+T2
6700  CONTINUE
      C1=(KF(I)/(TF(I)*TF(I)))*AINT2(NZWGL)
      XBAR(I+4)=XIC(I+4)+C1
      DO 6800 IT=1,NZWGL
      T1=-C1+(KF(I)/(TF(I)*TF(I)))*AINT2(IT)
      XWGLS(I+4,IT)=T1
6800  CONTINUE
6900  CONTINUE
      DO 7000 I=1,7
      TYPE *,XBAR(I)
7000  CONTINUE
      PAUSE 'CORIC'
      DO 7100 I=1,13
      ZBAR(I)=ZBARS(I,1)
7100  CONTINUE
      PRMT(1)=0.0
      PRMT(2)=TMAX
      PRMT(3)=HSLOW
      PRMT(4)=0.001
      DO 7200 I=1,7
      XDOT(I)=EWS
7200  CONTINUE
      ICRCT=0
      DO 7220 I=1,NXBR
      TDOT(I)=TBAR(I)
7220  CONTINUE
      NM1=NXBR-1
      DO 7260 I=1,NM1
      DT=TBAR(I+1)-TBAR(I)
      DO 7240 J=1,13
```

```
          ZDOT(J,I)=(ZBARS(J,I+1)-ZBARS(J,I))/DT
724Ø      CONTINUE
726Ø      CONTINUE
          DO 728Ø I=1,13
          ZDOT(I,NXBR)=ZDOT(I,NM1)
728Ø      CONTINUE
          DO 729Ø I=1,13
          TYPE *,ZDOT(I,1)
729Ø      CONTINUE
          PAUSE 'ZDOT'
          NDOT=NXBR
          NXBR=1
          CALL RKGS(PRMT,XBAR,XDOT,7,IHLF,FXBAR,OXBAR,AUX)
          NXBR=NXBR-1
          DO 74ØØ IT=1,NZWGL
          CALL INTPOL(TWGL(IT),TBAR,NXBR,JF,TINT)
          DO 73ØØ I=1,7
          XC(I,IT)=TINT*(XBARS(I,JF+1)-XBARS(I,JF))+XBARS(I,JF)
          1+XWGLS(I,IT)
73ØØ      CONTINUE
74ØØ      CONTINUE
          NXC=NZWGL
          DO 75ØØ I=1,NZWGL
          TXC(I)=TWGL(I)
75ØØ      CONTINUE
          PRMT(1)=Ø.Ø
          PRMT(2)=TMAX
          PRMT(3)=HFAST
          PRMT(4)=Ø.ØØ1
          DO 76ØØ I=1,13
          XDOT(I)=EWF
          ZWGL(I)=XIC(I+7)-ZBARS(I,1)
76ØØ      CONTINUE
          NZWGL=1
          ICRCT=1
          CALL RKGS(PRMT,ZWGL,XDOT,13,IHLF,FZWGL,OZWGL,AUX)
          NZWGL=NZWGL-1
          TYPE 77ØØ
77ØØ      FORMAT(' PLOT CORRECTED RESULTS? ',$)
78ØØ      ACCEPT 1,IANS
          IF(IANS-'Y') 79ØØ,81ØØ,79ØØ
79ØØ      IF(IANS-'N') 8ØØØ,82ØØ,8ØØØ
8ØØØ      TYPE 16ØØ,IANS
          GO TO 78ØØ
81ØØ      CALL GRAPHS(NXCT,NXC,NXBR,NZWGL,IBAUD,XS,TS,XC,TXC,
          1ZBARS,TBAR,ZWGLS,TWGL)
82ØØ      TYPE 83ØØ
83ØØ      FORMAT(' DO MORE CORRECTIONS? ',$)
84ØØ      ACCEPT 1,IANS
          IF(IANS-'Y') 85ØØ,48ØØ,85ØØ
85ØØ      IF(IANS-'N') 86ØØ,87ØØ,86ØØ
86ØØ      TYPE 16ØØ,IANS
          GO TO 84ØØ
87ØØ      TYPE 88ØØ
```

```
8500    FORMAT(' RUN ANOTHER CASE? ',S)
8900    ACCEPT 1,IANS
        IF(IANS-'Y') 9000,100,9000
9000    IF(IANS-'N') 9100,9200,9100
9100    TYPE 1600,IANS
        GO TO 8900
9200    STOP
        END
        !*************BLOCK DATA PAGE 2*************
        BLOCK DATA
        REAL KA(3),KE(3),KF(3)
        DIMENSION XD(3),XQ(3),XP(3),TDOP(3),TQOP(3),TA(3),TE(3),
        1TF(3),H(3),D(3)
        COMMON/BLK1/XD,XQ,XP
        COMMON/BLK2/TDOP,TQOP
        COMMON/BLK3/TA,TE,TF
        COMMON/BLK4/H,D
        COMMON/BLK5/ASAT,BSAT
        COMMON/BLK6/KA,KE,KF
        DATA (XD(I),I=1,3)/0.6,0.8958,0.9/
        DATA (XQ(I),I=1,3)/0.58,0.8645,0.85/
        DATA (XP(I),I=1,3)/0.0608,0.1198,0.1813/
        DATA (TDOP(I),I=1,3)/4.0,6.0,5.0/
        DATA (TQOP(I),I=1,3)/0.25,0.54,0.65/
        DATA (TA(I),I=1,3)/3*0.06/
        DATA (TE(I),I=1,3)/3*0.5/
        DATA (TF(I),I=1,3)/3*1.0/
        DATA (KA(I),I=1,3)/3*25.0/
        DATA (KE(I),I=1,3)/3*-0.0445/
        DATA (KF(I),I=1,3)/3*0.16/
        DATA (H(I),I=1,3)/23.64,6.4,3.01/
        DATA (D(I),I=1,3)/9.6,2.5,1.0/
        DATA ASAT/0.001123/,BSAT/0.3043/
        END
        !*************FUNCTION SE PAGE 3************
        FUNCTION SE(ARG)
        COMMON/BLK5/ASAT,BSAT
        SE=ASAT*EXP(BSAT*ARG)
        RETURN
        END
        !***********SUBROUTINE FXCT PAGE 4**********
        SUBROUTINE FXCT(TIME,X,XDOT)
        REAL KA(3),KE(3),KF(3)
        REAL ID(3),IQ(3)
        DIMENSION X(20),XDOT(20),XD(3),XQ(3),XP(3),TDOP(3),TQOP
        1(3),TA(3),TE(3),TF(3),H(3),D(3),Y(3,3),TH(3,3),VREF(3),
        2PIN(3),DELR(3),VLT(3)
        COMMON/BLK1/XD,XQ,XP
        COMMON/BLK2/TDOP,TQOP
        COMMON/BLK3/TA,TE,TF
        COMMON/BLK4/H,D
        COMMON/BLK6/KA,KE,KF
        COMMON/BLK7/Y,TH
        COMMON/BLK8/BH,SH
```

```
          COMMON/BLK9/VREF,PIN
          COMMON/BLK11/VLT
          DELR(1)=0.0
          DELR(2)=BH*X(8)/H(1)+H(3)*X(10)/SH
          DELR(3)=BH*X(8)/H(1)-H(2)*X(10)/SH
          OM1=X(1)-SH*X(9)/H(1)
          OM2=X(1)+X(9)+H(3)*X(11)/SH
          OM3=X(1)+X(9)-H(2)*X(11)/SH
          DO 200 I=1,3
          IU(I)=0.0
          IQ(I)=0.0
          DU 100 J=1,3
          ANG=TH(I,J)+DELR(J)-DELR(I)
          CANG=COS(ANG)
          SANG=SIN(ANG)
          IU(I)=ID(I)+Y(T,J)*(X(J+17)*CANG-X(J+1)*SANG)
          IQ(I)=IQ(I)+Y(I,J)*(X(J+1)*CANG+X(J+17)*SANG)
100       CONTINUE
200       CONTINUE
          PWRIN=PIN(1)/OM1+PIN(2)/OM2+PIN(3)/OM3
          DAM=D(1)*(OM1-1.0)+D(2)*(OM2-1.0)+D(3)*(OM3-1.0)
          PWROUT=0.0
          DU 300 I=1,3
          PWROUT=PWROUT+X(I+1)*IQ(I)+X(I+17)*ID(T)
300       CONTINUE
          XUOT(1)=(0.5/BH)*(PWRIN-DAM-PWROUT)
          DO 400 I=1,3
          XDOT(I+1)=(1.0/TDOP(I))*(-X(I+1)-(XD(I)-XP(I))*ID(I)+X(I
          1+11))
400       CONTINUE
          DO 500 I=1,3
          XDOT(I+4)=(1.0/TF(I))*(-X(I+4)+KF(I)*X(I+11)/TF(I))
500       CONTINUE
          XDOT(8)=377.0*X(9)
          XUOT(9)=(0.5/BH)*(-PIN(1)/OM1+D(1)*(OM1-1.0)+X(2)*IQ(1)+
          1X(18)*ID(1)+(H(1)/SH)*(PIN(2)/OM2-D(2)*(OM2-1.0)-X(3)*
          2IQ(2)-X(19)*ID(2)+PIN(3)/OM3-D(3)*(OM3-1.0)-X(4)*IQ(3)
          3-X(20)*ID(3)))
          XDOT(10)=377.0*X(11)
          XUOT(11)=(0.5/H(2))*(PIN(2)/OM2-D(2)*(OM2-1.0)-X(3)*IQ
          1(2)-X(19)*ID(2))-(0.5/H(3))*(PIN(3)/OM3-D(3)*(OM3-1.0
          2)-X(4)*IQ(3)-X(20)*ID(3))
          DO 600 I=1,3
          XDOT(I+11)=(1.0/TE(I))*(-(KE(I)+SE(X(I+11)))*X(I+11)+
          1X(I+14))
600       CONTINUE
          DO 700 I=1,3
          A1=X(I+17)+XP(I)*IQ(I)
          A2=X(I+1)-XP(I)*ID(I)
          VLT(I)=SQRT(A1*A1+A2*A2)
          XDOT(I+14)=(1.0/TA(I))*(KA(I)*(X(I+4)-KF(I)*X(I+11)/
          1TF(I)-VLT(I)+VREF(I))-X(I+14))
700       CONTINUE
          DO 800 I=1,3
```

```
        XDOT(I+17)=(1.0/TQOP(I))*(-X(I+17)+(XQ(I)-XP(I))*IQ(I))
800     CONTINUE
        RETURN
        END
        !************SUBROUTINE OXCT PAGE 5***********
        SUBROUTINE OXCT(TIME,X,XDOT,IHLF,NDIM,PRMT)
        DIMENSION X(20),XDOT(20),PRMT(5),XS(23,501),TS(501),
        1VLT(3)
        COMMON/BLK10/XS,TS,NXCT
        COMMON/BLK11/VLT
        DO 100 I=1,20
        XS(I,NXCT)=X(I)
100     CONTINUE
        DO 200 I=1,3
        XS(I+20,NXCT)=VLT(I)
200     CONTINUE
        TS(NXCT)=TIME
        IF(NXCT.GE.501) PRMT(5)=1.0
        NXCT=NXCT+1
        RETURN
        END
        !************SUBROUTINE SLVFST PAGE 6***********
        SUBROUTINE SLVFST(XBAR,ZBAR,TIME)
        DIMENSION H(3),D(3),XBAR(7),ZBAR(13),Y(3,3),TH(3,3),
        1TA(3),TE(3),TF(3),XD(3),XQ(3),XP(3),DELP(3),A(5,5),
        2B(5),WKAREA(5),V(3),PIN(3),VREF(3),ZDOT(13,501),TDOT(
        3501),DOT(13),TDOP(3),TQOP(3)
        REAL KA(3),KE(3),KF(3),ID(3),IQ(3)
        COMMON/BLK1/XD,XQ,XP
        COMMON/BLK2/TDOP,TQOP
        COMMON/BLK3/TA,TE,TF
        COMMON/BLK4/H,D
        COMMON/BLK5/ASAT,BSAT
        COMMON/BLK6/KA,KE,KF
        COMMON/BLK7/Y,TH
        COMMON/BLK8/RH,SH
        COMMON/BLK9/VREF,PIN
        COMMON/BLK15/ICRCT
        COMMON/BLK17/ZDOT,TDOT,NDOT
        C1=(1.0+H(1)/SH)*SIN(TH(1,2))
        C2=(1.0-H(1)/SH)*COS(TH(1,2))
        YH12=Y(1,2)*SQRT(C1*C1+C2*C2)
        PSI12=ATAN2(C1,C2)
        C1=(1.0+H(1)/SH)*SIN(TH(1,3))
        C2=(1.0-H(1)/SH)*COS(TH(1,3))
        YH13=Y(1,3)*SQRT(C1*C1+C2*C2)
        PSI13=ATAN2(C1,C2)
        C1=(1.0/H(2)+1.0/H(3))*SIN(TH(2,3))
        C2=(-1.0/H(2)+1.0/H(3))*COS(TH(2,3))
        YH23=0.5*Y(2,3)*SQRT(C1*C1+C2*C2)
        PSI23=ATAN2(C1,C2)
        DO 20 I=1,13
        DOT(I)=0.0
20      CONTINUE
```

```
      IF(ICRCT) 80,80,40
40    CALL INTPOL(TIME,TOOT,NOOT,JF,TINT)
      DO 60 I=1,13
      OOT(I)=ZOOT(I,JF)
60    CONTINUE
80    ZBAR(2)=OOT(1)/377.0
      ZBAR(4)=OOT(3)/377.0
      OM1=XBAR(1)-SH*ZBAR(2)/H(1)
      OM2=XBAR(1)+ZBAR(2)+H(3)*ZBAR(4)/SH
      OM3=XBAR(1)+ZBAR(2)-H(2)*ZBAR(4)/SH
      DO 1400 ITER=1,50
      ICONV=1
      DO 200 I=1,5
      B(I)=0.0
      DO 100 J=1,5
      A(I,J)=0.0
100   CONTINUE
200   CONTINUE
      ANG1=BH*ZBAR(1)/H(1)+H(3)*ZBAR(3)/SH
      ANG2=BH*ZBAR(1)/H(1)-H(2)*ZBAR(3)/SH
      GAM1=PSI12+ANG1
      GAM2=PSI13+ANG2
      T1=(H(1)/SH)*(PIN(2)/OM2+PIN(3)/OM3)-PIN(1)/OM1
      T2=O(1)*(OM1-1.0)-(H(1)/SH)*(O(2)*(OM2-1.0)+O(3)*(OM3
     1-1.0))
      T3=(XBAR(2)*XBAR(2)+ZBAR(11)*ZBAR(11))*Y(1,1)*COS
     1(TH(1,1))-(H(1)/SH)*((XBAR(3)*XBAR(3)+ZBAR(12)*ZBA
     2R(12))*Y(2,2)*COS(TH(2,2))+(XBAR(4)*XBAR(4)+ZBAR(13
     3)*ZBAR(13))*Y(3,3)*COS(TH(3,3)))
      P1=XBAR(2)*XBAR(3)+ZBAR(11)*ZBAR(12)
      P2=XBAR(2)*ZBAR(12)-XBAR(3)*ZBAR(11)
      P3=XBAR(2)*XBAR(4)+ZBAR(11)*ZBAR(13)
      P4=XBAR(2)*ZBAR(13)-XBAR(4)*ZBAR(11)
      P5=XBAR(3)*ZBAR(13)-XBAR(4)*ZBAR(12)
      P6=XBAR(3)*XBAR(4)+ZBAR(12)*ZBAR(13)
      T4=YH12*(P1*COS(GAM1)+P2*SIN(GAM1))
      T5=YH13*(P3*COS(GAM2)+P4*SIN(GAM2))
      T6=2.0*(H(1)/SH)*Y(2,3)*COS(TH(2,3))*(P5*SIN(ZBAR(3))
     1-P6*COS(ZBAR(3)))
      B(1)=-(T1+T2+T3+T4+T5+T6)+2.0*BH*OOT(2)
      T1=2.0*ZBAR(11)*Y(1,1)*COS(TH(1,1))
      T2=YH12*(ZBAR(12)*COS(GAM1)-XBAR(3)*SIN(GAM1))
      T3=YH13*(ZBAR(13)*COS(GAM2)-XBAR(4)*SIN(GAM2))
      A(1,1)=T1+T2+T3
      T1=-2.0*(H(1)/SH)*ZBAR(12)*Y(2,2)*COS(TH(2,2))
      T2=YH12*(ZBAR(11)*COS(GAM1)+XBAR(2)*SIN(GAM1))
      T3=-2.0*(H(1)/SH)*Y(2,3)*COS(TH(2,3))*(XBAR(4)*SIN(
     1ZBAR(3))+ZBAR(13)*COS(ZBAR(3)))
      A(1,2)=T1+T2+T3
      T1=-2.0*(H(1)/SH)*ZBAR(13)*Y(3,3)*COS(TH(3,3))
      T2=YH13*(ZBAR(11)*COS(GAM2)+XBAR(2)*SIN(GAM2))
      T3=2.0*(H(1)/SH)*Y(2,3)*COS(TH(2,3))*(XBAR(3)*SIN(
     1ZBAR(3))-ZBAR(12)*COS(ZBAR(3)))
      A(1,3)=T1+T2+T3
```

```
T1=(YH12*BH/H(1))*(-P1*SIN(GAM1)+P2*COS(GAM1))
T2=(YH13*BH/H(1))*(-P3*SIN(GAM2)+P4*COS(GAM2))
A(1,4)=T1+T2
T1=(YH12*H(3)/SH)*(-P1*SIN(GAM1)+P2*COS(GAM1))
T2=(YH13*H(2)/SH)*(P3*SIN(GAM2)-P4*COS(GAM2))
T3=(2.0*H(1)/SH)*Y(2,3)*COS(TH(2,3))*(P5*COS(ZBAR(3
1))+P6*SIN(ZBAR(3)))
A(1,5)=T1+T2+T3
GAM1=TH(2,1)-ANG1
GAM2=TH(3,1)-ANG2
GAM3=PSI23+ZBAR(3)
T1=0.5*(PIN(2)/(OM2*H(2))-PIN(3)/(OM3*H(3)))
T2=0.5*(-D(2)*(OM2-1.0)/H(2)+D(3)*(OM3-1.0)/H(3))
T3=0.5*(-(XBAR(3)*XBAR(3)+ZBAR(12)*ZBAR(12))*Y(2,2)*
1COS(TH(2,2))/H(2)+(XBAR(4)*XBAR(4)+ZBAR(13)*ZBAR(13)
2)*Y(3,3)*COS(TH(3,3))/H(3))
T4=-(0.5*Y(2,1)/H(2))*(P1*COS(GAM1)-P2*SIN(GAM1))
T5=(0.5*Y(3,1)/H(3))*(P3*COS(GAM2)-P4*SIN(GAM2))
T6=YH23*(P6*COS(GAM3)-P5*SIN(GAM3))
B(2)=-(T1+T2+T3+T4+T5+T6)+DOT(4)
T1=-0.5*(Y(2,1)/H(2))*(ZBAR(12)*COS(GAM1)+XBAR(3)*SIN(
1GAM1))
T2=0.5*(Y(3,1)/H(3))*(ZBAR(13)*COS(GAM2)+XBAR(4)*SIN(
1GAM2))
A(2,1)=T1+T2
T1=-ZBAR(12)*Y(2,2)*COS(TH(2.2))/H(2)
T2=-(0.5*Y(2,1)/H(2))*(ZBAR(11)*COS(GAM1)-XBAR(2)*SIN(
1GAM1))
T3=YH23*(ZBAR(13)*COS(GAM3)+XBAR(4)*SIN(GAM3))
A(2,2)=T1+T2+T3
T1=ZBAR(13)*Y(3,3)*COS(TH(3,3))/H(3)
T2=(0.5*Y(3,1)/H(3))*(ZBAR(11)*COS(GAM2)-XBAR(2)*SIN(
1GAM2))
T3=YH23*(ZBAR(12)*COS(GAM3)-XBAR(3)*SIN(GAM3))
A(2,3)=T1+T2+T3
T1=(0.5*Y(2,1)*BH/(H(2)*H(1)))*(-P1*SIN(GAM1)-P2*COS(
1GAM1))
T2=-(0.5*Y(3,1)*BH/(H(3)*H(1)))*(-P3*SIN(GAM2)-P4*COS(
1GAM2))
A(2,4)=T1+T2
T1=(0.5*Y(2,1)*H(3)/(H(2)*SH))*(-P1*SIN(GAM1)-P2*COS(
1GAM1))
T2=(0.5*Y(3,1)*H(2)/(H(3)*SH))*(-P3*SIN(GAM2)-P4*COS(
1GAM2))
T3=YH23*(-P6*SIN(GAM3)-P5*COS(GAM3))
A(2,5)=T1+T2+T3
DELR(1)=0.0
DELR(2)=ANG1
DELR(3)=ANG2
DO 400 I=1,3
IQ(I)=0.0
DO 300 J=1,3
ANG=TH(I,J)+DELR(J)-DELR(I)
CANG=COS(ANG)
```

```
        SANG=SIN(ANG)
        IQ(I)=IQ(I)+Y(I,J)*(XBAR(J+1)*CANG+ZBAR(J+1Ø)*SANG)
3ØØ     CUNTINUE
4ØØ     CUNTINUE
        DO 5ØØ I=1,3
        B(I+2)=ZBAR(I+1Ø)-(XQ(I)-XP(I))*IQ(I)+TQOP(I)*DOT(I+1Ø)
5ØØ     CONTINUE
        DO 9ØØ I=1,3
        DO 8ØØ J=1,3
        ANG=TH(I,J)+DELR(J)-DELR(I)
        IF(I-J) 7ØØ,6ØØ,7ØØ
6ØØ     A(I+2,J)=-(1.Ø-(XG(I)-XP(I))*Y(I,I)*SIN(TH(I,I)))
        GC TO 4ØØ
7ØØ     A(I+2,J)=(XG(I)-XP(I))*Y(I,J)*SIN(ANG)
8ØØ     CUNTINUE
9ØØ     CONTINUE
        GAM1=TH(1,2)+ANG1
        GAM2=TH(1,3)+ANG2
        T1=Y(1,2)*(-XBAR(3)*SIN(GAM1)+ZBAR(12)*COS(GAM1))
        T2=Y(1,3)*(-XBAR(4)*SIN(GAM2)+ZBAR(13)*COS(GAM2))
        A(3,4)=(XQ(1)-XP(1))*(BH/H(1))*(T1+T2)
        T1=H(3)*Y(1,2)*(-XBAR(3)*SIN(GAM1)+ZBAR(12)*COS(GAM1))
        T2=H(2)*Y(1,3)*(-XBAR(4)*SIN(GAM2)+ZBAR(13)*COS(GAM2))
        A(3,5)=((XQ(1)-XP(1))/SH)*(T1-T2)
        GAM1=TH(1,2)-ANG1
        GAM2=TH(2,3)-ZBAR(3)
        A(4,4)=-(XQ(2)-XP(2))*Y(2,1)*(BH/H(1))*(-XBAR(2)*SIN(GAM1)+
        1ZBAR(11)*COS(GAM1))
        T1=Y(2,1)*(H(3)/SH)*(-XBAR(2)*SIN(GAM1)+ZBAR(11)*COS(GAM1))
        T2=Y(2,3)*(-XBAR(4)*SIN(GAM2)+ZBAR(13)*COS(GAM2))
        A(4,5)=-(XQ(2)-XP(2))*(T1+T2)
        GAM1=TH(1,3)-ANG2
        GAM2=TH(2,3)+ZBAR(3)
        A(5,4)=-(XQ(3)-XP(3))*Y(1,3)*(BH/H(1))*(-XBAR(2)*SIN(GAM1)
        1+ZBAR(11)*COS(GAM1))
        T1=(Y(1,3)*H(2)/SH)*(-XBAR(2)*SIN(GAM1)+ZBAR(11)*COS(GAM1))
        T2=Y(2,3)*(-XBAR(3)*SIN(GAM2)+ZBAR(12)*COS(GAM2))
        A(5,5)=(XQ(3)-XP(3))*(T1+T2)
96Ø     CALL LEQT1F(A,1,5,5,B,Ø,WKAREA,IER)
        DO 12ØØ I=1,5
        IF(ABS(B(I))-Ø.ØØØ1) 12ØØ,12ØØ,11ØØ
11ØØ    ICONV=Ø
12ØØ    CONTINUE
        DO 13ØØ I=1,3
        ZBAR(I+1Ø)=B(I)+ZBAR(I+1Ø)
13ØØ    CONTINUE
        ZBAR(1)=ZBAR(1)+B(4)
        ZBAR(3)=ZBAR(3)+B(5)
        IF(ICONV) 14ØØ,14ØØ,16ØØ
14ØØ    CONTINUE
        TYPE 15ØØ
15ØØ    FORMAT(' NON-CONVERGENCE-ANGLES AND EDP')
        STOP
16ØØ    DELR(1)=Ø.Ø
```

```
         DELR(2)=RH*ZBAR(1)/H(1)+H(3)*ZBAR(3)/SH
         DELR(3)=RH*ZBAR(1)/H(1)-H(2)*ZBAR(3)/SH
         DO 1800 I=1,3
         ID(I)=0.0
         IQ(I)=0.0
         DO 1700 J=1,3
         ANG=TH(I,J)+DELR(J)-DELR(I)
         CANG=COS(ANG)
         SANG=SIN(ANG)
         ID(I)=ID(I)+Y(I,J)*(ZBAR(J+10)*CANG-XBAR(J+1)*SANG)
         IQ(I)=IQ(I)+Y(I,J)*(XBAR(J+1)*CANG+ZBAR(J+10)*SANG)
1700     CONTINUE
1800     CONTINUE
         DO 1900 I=1,3
         VD=ZBAR(I+10)+XP(I)*IQ(I)
         VQ=XBAR(I+1)-XP(I)*ID(I)
         V(I)=SORT(VD*VD+VQ*VQ)
1900     CONTINUE
         DO 2500 I=1,3
         DO 2200 ITER=1,50
         ICONV=1
         ANUM=(KE(I)+(KA(I)*KF(I)/TF(I))+SE(ZBAR(I+4)))*ZBAR
         1(I+4)-KA(I)*(XBAR(I+4)-V(I)+VREF(I))+TE(I)*DOT(I+4)+TA(
         1I)*DOT(I+7)
         DENOM=-(KE(I)+(KA(I)*KF(I)/TF(I))+(BSAT*ZBAR(I+4)+1.0
         1)*SE(ZBAR(I+4)))
         DIT=ANUM/DENOM
1960     IF(ABS(DIT)-0.0001) 2100,2100,2000
2000     ICONV=0
2100     ZBAR(I+4)=ZBAR(I+4)+DIT
         IF(ICONV)2200,2200,2400
2200     CONTINUE
         TYPE 2300,I
2300     FORMAT(' CONVERGENCE FAILURE-EFD ',I1)
         STOP
2400     ZBAR(I+7)=KA(I)*(XBAR(I+4)-KF(I)*ZBAR(I+4)/TF(I)-V(I)
         1+VREF(I))-TA(I)*DOT(I+7)
2500     CONTINUE
2600     RETURN
         ENO
         !************SUBROUTINE FXBAR PAGE 7**********
         SUBROUTINE FXBAR(TIME,XBAR,XDOT)
         REAL KA(3),KE(3),KF(3),ID(3),IQ(3)
         DIMENSION XBAR(7),XDOT(7),ZBAR(13),XD(3),XQ(3),XP(3),
         1TDOP(3),TQOP(3),TA(3),TE(3),TF(3),H(3),D(3),Y(3,3),
         2TH(3,3),VREF(3),PIN(3),DELR(3)
         COMMON/BLK1/XD,XQ,XP
         COMMON/BLK2/TDOP,TQOP
         COMMON/BLK3/TA,TE,TF
         COMMON/BLK4/H,D
         COMMON/BLK6/KA,KE,KF
         COMMON/BLK7/Y,TH
         COMMON/3LK8/RH,SH
         COMMON/BLK9/VREF,PIN
```

```
        COMMON/BLK12/ZBAR
        CALL SLVFST(XBAR,ZBAR,TIME)
        DELR(1)=0.0
        DELR(2)=BH*ZBAR(1)/H(1)+H(3)*ZBAR(3)/SH
        DELR(3)=BH*ZBAR(1)/H(1)-H(2)*ZBAR(3)/SH
        OM1=XBAR(1)-SH*ZBAR(2)/H(1)
        OM2=XBAR(1)+ZBAR(2)+H(3)*ZBAR(4)/SH
        OM3=XBAR(1)+ZBAR(2)-H(2)*ZBAR(4)/SH
        DU 200 I=1,3
        IO(I)=0.0
        IQ(I)=0.0
        DU 100 J=1,3
        ANG=TH(I,J)+DELR(J)-DELR(I)
        CANG=COS(ANG)
        SANG=SIN(ANG)
        IO(I)=IO(I)+Y(I,J)*(ZBAR(J+10)*CANG-YBAR(J+1)*SANG)
        IQ(I)=IQ(I)+Y(I,J)*(XBAR(J+1)*CANG+ZBAR(J+10)*SANG)
100     CONTINUE
200     CONTINUE
        PWRIN=PIN(1)/OM1+PIN(2)/OM2+PIN(3)/OM3
        DAM=D(1)*(OM1-1.0)+D(2)*(OM2-1.0)+D(3)*(OM3-1.0)
        PWROUT=0.0
        DU 400 I=1,3
        PWROUT=PWROUT+XBAR(I+1)*IQ(I)+ZBAR(I+10)*ID(I)
400     CONTINUE
        XUOT(1)=(0.5/BH)*(PWRIN-DAM-PWROUT)
        DO 500 I=1,3
        XDOT(I+1)=(1.0/TDOP(I))*(-XBAR(I+1)-(XD(I)-XP(I))*
        1ID(I)+ZBAR(I+4))
500     CONTINUE
        DO 600 I=1,3
        XDOT(I+4)=(1.0/TF(I))*(-XBAR(I+4)+KF(I)*ZBAR(I+4)
        1/TF(I))
600     CONTINUE
        RETURN
        END
        !***********SUBROUTINE OXBAR PAGE 8***********
        SUBROUTINE OXBAR(TIME,XBAR,XDOT,IHLF,NDIM,PRMT)
        DIMENSION XBAR(7),XDOT(7),PRMT(5),XBARS(7,501),ZBARS(
        113,501),ZBAR(13),TBAR(501)
        COMMON/BLK12/ZBAR
        COMMON/BLK13/XBARS,ZBARS,TBAR,NXBR
        DO 100 I=1,7
        XBARS(I,NXBR)=XBAR(I)
100     CONTINUE
        DO 200 I=1,13
        ZBARS(I,NXBR)=ZBAR(I)
200     CONTINUE
        TBAR(NXBR)=TIME
        IF(NXBR.GE.501) PRMT(5)=1.0
        NXBR=NXBR+1
        RETURN
        END
        !***********SUBROUTINE FZWGL PAGE 9***********
```

```
      SUBROUTINE FZWGL(TIME,ZWGL,XDOT)
      REAL KA(3),KE(3),KF(3),ID(3),IQ(3)
      DIMENSION ZWGL(13),XDOT(13),XD(3),XQ(3),XP(3),TDOP(3),
      1TQOP(3),TA(3),TE(3),TF(3),H(3),D(3),Y(3,3),TH(3,3),
      2VREF(3),PIN(3),DELR(3),XBARS(7,501),ZBARS(13,501),
      3TBAR(501),XC(7,501),TXC(501),X(7),Z(13)
      COMMON/BLK1/XD,XQ,XP
      COMMON/BLK2/TDOP,TQOP
      COMMON/BLK3/TA,TE,TF
      COMMON/BLK4/H,D
      COMMON/BLK6/KA,KE,KF
      COMMON/BLK7/Y,TH
      COMMON/BLK8/BH,SH
      COMMON/BLK9/VREF,PIN
      COMMON/BLK13/XBARS,ZBARS,TBAR,NXBR
      COMMON/BLK14/XC,TXC,NXC
      COMMON/BLK15/ICRCT
      IF(ICRCT) 100,100,300
100   CALL INTPOL(TIME,TBAR,NXBR,JFIND,TINT)
      DO 200 I=1,7
      X(I)=TINT*(XBARS(I,JFIND+1)-XBARS(I,JFIND))+XBARS(
      1I,JFIND)
200   CONTINUE
      GO TO 500
300   CALL INTPOL(TIME,TXC,NXC,JFIND,TINT)
      DO 400 I=1,7
      X(I)=TINT*(XC(I,JFIND+1)-XC(I,JFIND))+XC(I,JFIND)
400   CONTINUE
500   CALL INTPOL(TIME,TBAR,NXBR,JFIND,TINT)
      DO 600 I=1,13
      Z(I)=TINT*(ZBARS(I,JFIND+1)-ZBARS(I,JFIND))+ZBARS(
      1I,JFIND)+ZWGL(I)
600   CONTINUE
      DELR(1)=0.0
      DELR(2)=BH*Z(1)/H(1)+H(3)*Z(3)/SH
      DELR(3)=BH*Z(1)/H(1)-H(2)*Z(3)/SH
      OM1=X(1)-SH*Z(2)/H(1)
      OM2=X(1)+Z(2)+H(3)*Z(4)/SH
      OM3=X(1)+Z(2)-H(2)*Z(4)/SH
      DO 800 I=1,3
      ID(I)=0.0
      IQ(I)=0.0
      DO 700 J=1,3
      ANG=TH(I,J)+DELR(J)-DELR(I)
      CANG=COS(ANG)
      SANG=SIN(ANG)
      ID(I)=ID(I)+Y(I,J)*(Z(J+10)*CANG-X(J+1)*SANG)
      IQ(I)=IQ(I)+Y(I,J)*(X(J+1)*CANG+Z(J+10)*SANG)
700   CONTINUE
800   CONTINUE
      XDOT(1)=377.0*ZWGL(2)
      XDOT(2)=(0.5/BH)*(-PIN(1)/OM1+D(1)*(OM1-1.0)+X(2)*
      1IQ(1)+Z(11)*ID(1)+(H(1)/SH)*(PIN(2)/OM2-D(2)*(OM2
      2-1.0)-X(3)*IQ(2)-Z(12)*ID(2)+PIN(3)/OM3-D(3)*(OM3
      3-1.Ø)-X(4)*IQ(3)-Z(13)*ID(3)))
      XDOT(3)=377.Ø*ZWGL(4)
      XDOT(4)=(Ø.5/H(2))*(PIN(2)/OM2-D(2)*(OM2-1.Ø)-X(3)*
      1IQ(2)-Z(12)*ID(2))-(Ø.5/H(3))*(PIN(3)/OM3-D(3)*(
      2OM3-1.Ø)-X(4)*IQ(3)-Z(13)*ID(3))
      DO 9ØØ I=1,3
      XDOT(I+4)=(1.Ø/TE(I))*(-(KE(I)+SE(Z(I+4)))*Z(I+4)+
      1Z(I+7))
9ØØ   CONTINUE
      DO 1ØØØ I=1,3
      A1=Z(I+1Ø)+XP(I)*IQ(I)
      A2=X(I+1)-XP(I)*ID(I)
      VLT=SQRT(A1*A1+A2*A2)
      XDOT(I+7)=(1.Ø/TA(I))*(KA(I)*(X(I+4)-KF(I)*Z(I+4)
      1/TF(I)-VLT+VREF(I))-Z(I+7))
1ØØØ  CONTINUE
      DO 11ØØ I=1,3
      XDOT(I+1Ø)=(1.Ø/TQOP(I))*(-Z(I+1Ø)+(XQ(I)-XP(I))*IQ
      1(I))
11ØØ  CONTINUE
      PETURN
      END
      !***********SUBROUTINE OZWGL PAGE 1Ø**********
      SUBROUTINE OZWGL(TIME,ZWGL,XDOT,IHLF,NDIM,PRMT)
      DIMENSION ZWGL(13),XDOT(13),PRMT(5),ZWGLS(13,5Ø1),
      1TWGL(5Ø1)
      COMMON/BLK16/ZWGLS,TWGL,NZWGL
      DO 1ØØ I=1,13
      ZWGLS(I,NZWGL)=ZWGL(I)
1ØØ   CONTINUE
      TWGL(NZWGL)=TIME
      IF(NZWGL.GE.5Ø1) PRMT(5)=1.Ø
      NZWGL=NZWGL+1
      RETURN
      END
      !***********SUBROUTINE INTPOL PAGE 11***********
      SUBROUTINE INTPOL(TIME,T,N,JFIND,TINT)
      DIMENSION T(N)
      DO 1ØØ JT=1,N
      JFIND=JT
      IF(ABS(TIME-T(JT))-1.ØE-Ø5) 2ØØ,2ØØ,1ØØ
1ØØ   CONTINUE
      GO TO 3ØØ
2ØØ   TINT=Ø.Ø
      RETURN
3ØØ   NM1=N-1
      DO 6ØØ JT=1,NM1
      JFIND=JT
      IF(TIME-T(JT)) 6ØØ,6ØØ,4ØØ
4ØØ   IF(TIME-T(JT+1))5ØØ,6ØØ,6ØØ
5ØØ   TNUM=TIME-T(JT)
      TDENOM=T(JT+1)-T(JT)
      TINT=TNUM/TDENOM
      RETURN
```

```
600     CONTINUE
        DIFF=TIME-T(JFIND)
        TYPE 700,TIME,T(JFIND),DIFF
700     FORMAT(' ',3(2X,E15.8))
        TYPE 800
800     FORMAT(' ***INTERPOLATION FAILURE***')
        PAUSE 'INTPOL'
        STOP
        END
        !**********SUBROUTINE GRAPHS PAGE 12**********
        SUBROUTINE GRAPHS(NXCT,NXHAT,NZBAR,NZWGL,IBAUD,XS,TS,
        1XHAT,TXHAT,ZBAR,TZBAR,ZWGL,TZWGL)
        DIMENSION XS(23,NXCT),TS(NXCT),XHAT(7,NXHAT),TXHAT(
        1NXHAT),ZBAR(13,NZBAR),TZBAR(NZBAR),ZWGL(13,NZWGL),
        2TZWGL(NZWGL),PLT(501),ZHAT(501)
100     TYPE 200
200     FORMAT(' WHICH STATE WILL BE PLOTTED? ',$)
        ACCEPT 300,IPLT
300     FORMAT(I)
        IF(IPLT-7) 800,800,400
400     IF(IPLT-20) 1300,1300,500
500     DO 600 I=1,NXCT
        PLT(I)=XS(IPLT,I)
600     CONTINUE
        CALL INITT(IBAUD)
        CALL BINITT
        CALL NPTS(NXCT)
        CALL XFRM(4)
        CALL YFRM(4)
        CALL CHECK(TS,PLT)
        CALL DSPLAY(TS,PLT)
        CALL FRAME
        CALL ANMODE
        ACCEPT 700,IANS
700     FORMAT(A1)
        GO TO 1800
800     AMIN=1.0E30
        AMAX=-1.0E30
        DO 900 I=1,NXCT
        IF(AMIN.GT.XS(IPLT,I)) AMIN=XS(IPLT,I)
        IF(XS(IPLT,I).GT.AMAX) AMAX=XS(IPLT,I)
900     CONTINUE
        DO 1000 I=1,NXHAT
        IF(AMIN.GT.XHAT(IPLT,I)) AMIN=XHAT(IPLT,I)
        IF(XHAT(IPLT,I).GT.AMAX) AMAX=XHAT(IPLT,I)
1000    CONTINUE
        DO 1100 I=1,NXCT
        PLT(I)=XS(IPLT,I)
1100    CONTINUE
        CALL INITT(IBAUD)
        CALL BINITT
        CALL DLIMY(AMIN,AMAX)
        CALL NPTS(NXCT)
        CALL XFRM(4)
        CALL YFRM(4)
        CALL CHECK(TS,PLT)
        CALL DSPLAY(TS,PLT)
        CALL FRAME
        DO 1200 I=1,NXHAT
        PLT(I)=XHAT(IPLT,I)
1200    CONTINUE
        CALL LINE(34)
        CALL DLIMY(AMIN,AMAX)
        CALL NPTS(NXHAT)
        CALL CHECK(TXHAT,PLT)
        CALL CPLOT(TXHAT,PLT)
        CALL ANMODE
        ACCEPT 700,IANS
        GO TO 1800
1300    JPLT=IPLT-7
        DO 1400 I=1,NZWGL
        TIME=TZWGL(I)
        CALL INTPOL(TIME,TZBAR,NZBAR,JFIND,TINT)
        ZHAT(I)=TINT*(ZBAR(JPLT,JFIND+1)-ZBAR(JPLT,JFIND))
        1+ZBAR(JPLT,JFIND)+ZWGL(JPLT,I)
1400    CONTINUE
        AMIN=1.0E30
        AMAX=-1.0E30
        DO 1500 I=1,NXCT
        IF(AMIN.GT.XS(IPLT,I)) AMIN=XS(IPLT,I)
        IF(XS(IPLT,I).GT.AMAX) AMAX=XS(IPLT,I)
1500    CONTINUE
        DO 1600 I=1,NZWGL
        IF(AMIN.GT.ZHAT(I)) AMIN=ZHAT(I)
        IF(ZHAT(I).GT.AMAX) AMAX=ZHAT(I)
1600    CONTINUE
        DO 1700 I=1,NXCT
        PLT(I)=XS(IPLT,I)
1700    CONTINUE
        CALL INITT(IBAUD)
        CALL BINITT
        CALL DLIMY(AMIN,AMAX)
        CALL NPTS(NXCT)
        CALL XFRM(4)
        CALL YFRM(4)
        CALL CHECK(TS,PLT)
        CALL DSPLAY(TS,PLT)
        CALL FRAME
        CALL LINE(34)
        CALL DLIMY(AMIN,AMAX)
        CALL NPTS(NZWGL)
        CALL CHECK(TZWGL,ZHAT)
        CALL CPLOT(TZWGL,ZHAT)
        CALL ANMODE
        ACCEPT 700,IANS
1800    TYPE 1900
1900    FORMAT(' MORE PLOTS? ',$)
2000    ACCEPT 700,IANS
```

```
        IF(IANS-'Y') 2100,100,2100
2100    IF(IANS-'N') 2200,2400,2200
2200    TYPE 2300,IANS
2300    FORMAT(' ??',A1,'?? TYPE AGAIN ',$)
        GO TO 2000
2400    RETURN
        END
```

REFERENCES

1. E. W. Kimbark, Power System Stability, Vol. 1, John Wiley, New York, 1948.

2. J. E. Van Ness, Hewlon Zimmer, and Mehmet Cultu, "Reduction of Dynamic Models of Power Systems," 1973 PICA Proc., pp. 105-112, 1973.

3. J. M. Undrill and A. E. Turner, "Construction of Power System Electromechanical Equivalents by Modal Analysis," IEEE Trans. Power Apparatus and Systems, Vol. PAS-90, pp. 2049-2059, 1971.

4. J. M. Undrill, J. A. Casazza, E. M. Gulachenski, and L. K. Kirchmayer, "Electromechanical Equivalents for Use in Power System Stability Studies," IEEE Trans. Power Apparatus and Systems, Vol. PAS-90, pp. 2060-2071, 1971.

5. R. W. De Mello, R. Podmore, and K. N. Stanton, "Coherency Based Dynamic Equivalents for Transient Stability Studies," EPRI Final Report, 1974.

6. J. H. Chow, J. J. Allemong, and P. V. Kokotovic, "Singular Perturbation Analysis of Systems with Sustained High Frequency Oscillations," Automatica, Vol. 14, pp. 271-279, 1978.

7. R. P. Schulz, A. E. Turner, and D. N. Ewart, "Long Term Power System Dynamics," EPRI Final Report, 1974.

8. O. W. Hanson, C. J. Goodwin, and P. L. Dandeno, "Influence of Excitation and Speed Control Parameters in Stabilizing Intersystem Oscillations," IEEE Trans. Power Apparatus and Systems, Vol. PAS-87, pp. 1306-1313, 1968.

9. J. M. Undrill, "Equipment and Load Modeling in Power System Dynamic Simulation," Systems Engineering for Power: Status and Prospects, ERDA, pp. 394-420, 1975.

10. P. V. Kokotovic, R. E. O'Malley, Jr., and P. Sannuti, "Singular Perturbations and Order Reduction in Control Theory - An Overview," Automatica, Vol. 12, pp. 123-132, 1976.

11. K. W. Chang, "Singular Perturbations of a General Boundary Value Problem," SIAM J. Math. Anal., Vol. 3, pp. 520-526, 1972.

12. P. V. Kokotovic, "A Riccati Equation for Block-Diagonalization of Ill-Conditioned Systems," IEEE Trans. Automatic Control, Vol. AC-20, pp. 812-814, 1975.

13. IEEE Committee Report, "Computer Representation of Excitation Systems," IEEE Trans. Power Apparatus and Systems, Vol. PAS-87, pp. 1460-1464, 1968.

14. A. B. Vasil'eva, "Asymptotic Behavior of Solutions of Certain Problems for Ordinary Non-Linear Differential Equations with a Small Parameter Multiplying the Highest Derivatives," Usp. Mat. Nauk., Vol. 18, pp. 15-86, 1963.

15. K. N. Stanton, "Dynamic Energy Balance Studies for Simulation of Power-Frequency Transients," 1971 PICA Proc.,pp. 173-179, 1971.

16. E. W. Kimbark, Power System Stability, Vol. 3, John Wiley, New York, 1956.

17. IEEE Committee Report, "Recommended Phasor Diagram for Synchronous Machines," IEEE Trans. Power Apparatus and Systems, Vol. PAS-88, pp. 1593-1610, 1969.

18. G. Shackshaft, "General-Purpose Turbo-Alternator Model," Proc. IEE, Vol. 110, pp. 703-713, 1963.

19. P. L. Dandeno, P. Kundur, and R. P. Schulz, "Recent Trends and Progress in Synchronous Machine Modeling in the Electric Utility Industry," Proc. IEEE, Vol. 62, pp. 941-950, 1974.

20. D. N. Ewart and R. P. Schulz, "FACE Multi-Machine Power System Simulator Program," 1969 PICA Proc., pp. 133-153, 1969.

21. A. W. Rankin, "Per-Unit Impedances of Synchronous Machines," AIEE Trans., Vol. 64, pp. 569-573, 1945.

VITA

John Jay Allemong was born in Harvey, Illinois on January 23, 1951. He received the BSEE degree with Highest Honors and University Honors and the MSEE degree both from the University of Illinois, Urbana-Champaign in June 1973 and May 1974 respectively.

He was voted the Eta Kappa Nu Outstanding Senior in Electrical Engineering in 1973. From February 1973-February 1974 he held an RCA Fellowship in Electrical Engineering. From July 1974-August 1975 he was employed as an Electrical Engineer by the consulting firm of Sargent and Lundy Engineers, Chicago, Illinois.

Since August 1975 he has held the positions of Instructor in the Department of Electrical Engineering and Graduate Research Assistant in the Coordinated Science Laboratory of the University of Illinois. In April 1977 he received the Harold L. Olesen Award for excellence in undergraduate teaching by a graduate student. His interests are power systems and the application of computers to power systems.

Mr. Allemong is a student member of the Institute of Electrical and Electronics Engineers.